PCT/IB04/50633 NL030582

Europäisches Patentamt

European Patent Office

Office européen des brevets

PCT/IB04/50633

# Bescheinigung

# Certificate

# Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

**Patentanmeldung Nr.** **Patent application No.** **Demande de brevet n°**

03076440.1

PRIORITY DOCUMENT
SUBMITTED OR TRANSMITTED IN COMPLIANCE WITH RULE 17.1(a) OR (b)

REC'D 14 JUN 2004
WIPO PCT

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

R C van Dijk

Europäisches Patentamt
European Patent Office
Office européen des brevets

Anmeldung Nr:
Application no.: 03076440.1
Demande no:

Anmeldetag:
Date of filing: 12.05.03
Date de dépôt:

Anmelder/Applicant(s)/Demandeur(s):

Koninklijke Philips Electronics N.V.
Groenewoudseweg 1
5621 BA Eindhoven
PAYS-BAS

Bezeichnung der Erfindung/Title of the invention/Titre de l'invention:
(Falls die Bezeichnung der Erfindung nicht angegeben ist, siehe Beschreibung.
If no title is shown please refer to the description.
Si aucun titre n'est indiqué se referer à la description.)

Iterative stripewise trellis-based symbol detection method and device for multidimentional recording systems

In Anspruch genommene Prioriät(en) / Priority(ies) claimed /Priorité(s) revendiquée(s)
Staat/Tag/Aktenzeichen/State/Date/File no./Pays/Date/Numéro de dépôt:

Internationale Patentklassifikation/International Patent Classification/
Classification internationale des brevets:

G11B20/00

Am Anmeldetag benannte Vertragstaaten/Contracting states designated at date of filing/Etats contractants désignées lors du dépôt:

AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL
PT RO SE SI SK TR LI

# Iterative stripewise trellis-based symbol detection method and device for multi-dimensional recording systems

## FIELD OF THE INVENTION

The invention relates to a trellis-based symbol detection method for detecting symbols of a channel data block recorded on a record carrier. The invention applies to digital recording systems, such as magnetic recording and optical recording systems. It is particularly advantageous for two-dimensional optical recording, which is one of the potential technologies for the next generations of optical recording.

## BACKGROUND ART

Current state-of-the-art optical disc systems are based on one-dimensional (1D) Optical Recording. A single laser beam is directed at a single track of information, which forms a continuous spiral on the disc, spiraling outwards to the outer edge of the disc. The single spiral contains a single (or one dimensional, 1D) track of bits. The single track consists of sequences of very small pit-marks or pits and the spaces between them, which are called land-marks or lands. The laser light is diffracted at the pit structures of the track. The reflected light is detected on a photo-detector Integrated Circuit (IC), and a single high-frequency signal is generated, which is used as the waveform from which bit-decisions are derived. A new route for the 4th generation of optical recording technology that will succeed "Blue Ray Disc" also called "DVR" already succeeding DVD (Digital Video Disc) technology is based on two-dimensional (2D) binary optical recording. 2D recording means that e.g. 10 tracks are recorded in parallel on the disc without guard space in between. Then, the 10 tracks together form one big spiral. The format of a disc for 2D optical recording (called in short a "2D disc") is based on that broad spiral, in which the information is recorded in the form of 2D features. The information is written as a honeycomb structure and is encoded with a 2D channel code, which facilitates bitdetection. The disc shall be read out with an array of e.g. 10 (or more) optical spots, which are sampled in time, in order to obtain a two dimensional array of samples in the player. Parallel read out is realized using a single laser beam, which passes through a grating, which produces the array of laser spots. The array of spots scans the full width of the broad spiral. The light from each laser spot is reflected by the 2D pattern on the disc, and is detected on a photo-detector IC, which generates a number of highfrequency signal waveforms. The set of signal waveforms is used as the input of the 2D signal processing. The motivation behind 2D recording is that much less disc space is wasted as guard space, so that the recording capacity of the disc can be increased. Although 2D recording is first studied for optical recording, similarly, magnetic recording can also be made two-dimensional. One of the new aspects of such recording techniques is that they require two dimensional signal processing. In particular, one optical spot must be considered as a device which takes a plane of "pits"/"lands" (or "marks" and "non-marks") as input and produces a corresponding output. The optical spot transfer function has the characteristics of a 2D low pass filter, whose shape can be approximated by a cone. Apart from its linear transfer characteristics, the 2D optical channel also has non-linear contributions. The radius of the cone corresponds to the cutoff frequency, determined by the numerical aperture of the lens, and the wavelength of the light. This filtering characteristic causes 2D Inter Symbol Interference (ISI) in the player. It is the task of a bit-detector to annihilate (most of) this ISI (which can be both linear and non-linear). An optimal way to implement a bit-detector is to use a Viterbi algorithm. A Viterbi bitdetector does not

amplify the noise. If soft decision output, i.e. reliability information about the bits, is required, a dual -Viterbi i.e (Max-)(Log-)MAP, or MAP, or SOVA (Soft Output Viterbi) algorithm can be used. One of the difficulties of designing a bit-detector for the 2D case, is that a straightforward Viterbi bit-detector would need as its "state", one or more columns of "old" track bits because of the memory of the ISI. If e.g. 10 tracks are recorded in parallel in the 2D broad spiral, and e.g. two old bits per track is needed for a proper description of the state because of the tangential extent (along-the-tracks) of the 2D impulse response, this results in a state of 2 x 10 = 20 bits. Thus, the number of states in the Viterbi (or MAP, (Max-)(Log-)MAP, MAP, SOVA, etc.) algorithm becomes $2^{20}$, which is completely impractical. This requires a different approach, which may be slightly sub optimal, but has a significantly reduced complexity. By providing a method of stripe wise bit detection where the stripe wise detector uses side information from adjacent tracks the bit detection of a broad spiral can be segmented, reducing the complexity of the overall detection method. The use of side information however can introduced errors into the bit detection.

## SUMMARY OF THE INVENTION

It is an object of the invention to provide a symbol detection method that does not degrade in performance due to unreliable side information.

To achieve this objective the bit detection method is characterized in that a weighing of a contribution of the side information is assigned based on a reliability of the side information When ,due to the nature of the source of the side information, the side information is not reliable the contribution to the bit detection of this side information is reduced by applying a weighing factor in accordance with the reliability of the side information. The contribution of unreliable side information receives a lower weighing factor then the contribution of reliable side information.

An embodiment of the symbol detection method is characterized in that the contribution is a contribution to an objective function of the search based algorithm. An objective function of a search based algorithm is typically minimizing the error between the transmitted or recorded data and the detected data by searching among all candidate possibilities the most likely candidate. The contribution can be a contribution to a branch metric.

An embodiment of the symbol detection method is characterized in that the search based algorithm comprises the use of internal contributions and that the use of the internal contributions comprises assigning an individual weighing of the internal contributions In addition to weighing the contribution of the side information from outside the stripe the contributions from within the stripe can also be weighed to reduce contribution from unreliable parts of the stripe.
For instance the detection of bits in a row directly adjacent to a yet to be processed stripe is unreliable because all bits around a bit to be detected contribute to the detection of that bit but only the bits inside the stripe are contributing to the detection with their most likely values while the value of the bits outside the stripe but still contributing to the detection of that bit are less reliable and in a first iteration even unknown.
The weighing of the contribution of that bit, although inside the stripe, must be reduced in order to reduce the contribution from the unreliable bits just outside the stripe. If the bits outside the strip that contribute to the detection are not yet known, they can be assumed to all have the value 0, 1 or have a random value in order to have a value for the bit detection, even if that value is incorrect.

The bit being detected is thus less reliable and because the bit being detected is also used in the detection of its neighbors, those neighbors also receive contributions that are less reliable than desirable.
The weighing of the contribution of the bit to the detection of other bits in the stripe must thus also be reduced resulting in potentially different weighing of the contributions from inside the stripe.

An embodiment of the symbol detection method is characterized in that the search based algorithm is a Viterbi algorithm, a sequential decoding algorithm such as a stack algorithm or a Fano algorithm, or a soft-decision output algorithm such as a (Max)(Log)MAP algorithm, or a reduced complexity maximum likelihood detection algorithm.
The listed search based algorithms all can be used to perform bit detection, allow the introduction of contribution of side information and allow the weighing of contributions. They are there fore suitable algorithms that can be used with the bit detection method according to the invention.

An embodiment of the symbol detection method is characterized in that the side information is an estimated channel input symbol.
Hard decision bit detection methods produce side information in the form of an estimated channel input symbol. The contribution of the estimated channel input symbol is used during bit detection.

An embodiment of the symbol detection method is characterized in that the side information is likelihood information about a channel input symbol
Soft decision bit detection algorithms produce side information in the form of likelihood information about a channel input symbol. The likelihood information of the estimated input symbol is used during bit detection

An embodiment of the symbol detection method is characterized in that a further side information, derived from the row adjacent to the first stripe, is being used in the estimation of said symbol values.
Not only side information derived from channel input symbols can be used but also other side information derived from the adjacent stripe can be used. All side information derived from the adjacent stripe contribute to a more reliable bit detection.

An embodiment of the symbol detection method is characterized in that the further side information comprises channel output values.
Not only side information derived from channel input symbols can be used but also other side information derived from the channel output values of the adjacent stripe can be used. This additional side information when used in tandem with the side information derived from the estimated input symbols contributes to a more reliable bit detection.

An embodiment of the symbol detection method is characterized in that the channel output values are filtered channel output values.
Filtered output values are often readily available and can be used to derive side information from.

An embodiment of the symbol detection method is characterized in that a weighing of the contribution of the side information is highest for side information derived from a symbol detection with a highest reliability.
There can be multiple adjacent or overlapping stripes for a stripe to be processed. Each adjacent stripe then provides side information. In order to improve the bit detection of the stripe to be processed each contribution from the side information of the adjacent stripes is weighed and more reliable contribution are given a higher weight than less reliable contributions. This way the less reliable contributions contribute less to the bit detection resulting in a more reliable bit detection.

An embodiment of the symbol detection method is characterized in that the symbol detection with the highest reliability is a symbol detection from a previous iteration.
With every iteration the reliability of the side information derived from the bit detection of that iteration increases because the reliability of the overall bit detection increases with every iteration.
Thus the weightings can be increased from one iteration to the next to reflect this increased reliability of the side information.

An embodiment of the symbol detection method is characterized in that the weighing is based on a distance between a position of a symbol value to be detected and a position of a side information symbol position.
When side information is located further away from the position of the symbol value to be detected the contribution of that side information is less than when the side information is located close to the symbol value to be detected. The weighing reflects this reduced contribution. This assures that side information from further away from the symbol value to be detected contributes less to the symbol detection.

An embodiment of the symbol detection method is characterized in that the distance is a distance to a most reliable side information position and that the weighing is a highest weighing.
The side information derived from the closest distance has the largest contribution to the symbol detection.
The weighing reflects this contribution by assigning the highest weight to the contribution of this side information.

An embodiment of the symbol detection method is characterized in that the weighing of the contribution of the side information is different for the second detector compared to the first detector
When the stripe is processed by multiple bit detectors in parallel the weighing of contributions can differ from one detector to another detector, for instance because the reliability of the symbol detection from which the side information is derived varies from stripe to stripe across the broad spiral which the stripes together form.

An embodiment of the symbol detection method is characterized in that the weighing of the contribution of the side information is different for a second iteration compared to a first iteration.
When the stripe is processed in multiple iterations the weighing can be varied to reflect the increase or decrease in reliability of the side information from one iteration to the next iteration.

An embodiment of the symbol detection method is characterized in that the weighing of the contribution of the side information is higher for the second iteration compared to the first iteration.
In general the reliability of the symbol detection and thus of the side information increases from one iteration to the next iteration. The weighing can be adjusted to reflect this increase in reliability from one iteration to the next iteration.

An embodiment of the symbol detection method is characterized in that the side information is obtained from a row comprising data which is highly protected using redundant coding. When a row comprising data which is highly protected is comprised in the broad spiral or delimits the broad spiral the side information derived from this data is also more reliable than side information derived from the regular stripes. A higher weighing must therefore be assigned to the side information derived from data which is highly protected compared to side information derived from other data.

An embodiment of the symbol detection method is characterized in that the side information is obtained from a row comprising predefined data.
Predefined data is inherently reliable in detection because errors can be easily corrected. As a consequence side information derived from predefined data is also reliable. The weighing of side information derived from predefined data can thus be higher than the weighing of side information derived from other data.

An embodiment of the symbol detection method is characterized in that the row comprising data which is highly protected using redundant coding is a guard band.
A guard band often comprises predefined data or is highly protected in order to assure correct detection of the guard band fro purposes of tracking etc. The guard band can thus be put to dual use by deriving side information from the data in the guard band and providing this side information to the symbol detector of the stripe adjacent to the guard band to improve the reliability of the detection.

An embodiment of the symbol detection method is characterized in that the row comprising data which is highly protected using redundant coding is located centrally between the rows forming the set of symbol rows.
Typically a row comprising highly protected data is located such that it delimits the data area. It is however also possible to locate such a row in the middle of the data area. Such a highly protected row can be located in the data area at a position where the stripe wise detection is inherrently less reliable, for instance near the center of the data area. In the case of a broad spiral the row would be located near the middle of the broad spiral. Since the reliability of the row comprising highly protected propagates to the adjacent stripes which use, directly or indirectly, use side information from the row comprising highly protected data, such a row can be appropriately positioned to enhance the detection where needed.

An embodiment of the symbol detection method is characterized in that the N-Dimensional channel tube is delimited by multiple guard bands.
By using multiple guard bands the methods outlined in the previous embodiments can be used to start multiple bit detectors in parallel. Near each guard band a bit detector starts, using the side information derived from that guard band, a cascade of bit detectors where each bit detector in the cascade closely trails the previous detector in the cascade. When using the 2 dimensional broad spiral as an example there would be for instance 2 guard bands, a first guards band delimiting the broad spiral at the top and a second guard band delimiting the broad spiral at the bottom. A first cascade of bit detectors starts at the first

guard band and propagating the increased reliability down in the cascade towards the second guard band. A second cascade of bit detectors starts at the second guard band and propagating the increased reliability up in the cascade towards the first guard band.
The two cascades of bit detectors would meet somewhere on the broad spiral, for instance at the middle of the broad spiral, each having processed the upper portion of stripes of the broad spiral, respectively the lower portion of stripes of the broad spiral.
In a graphic sense the cascades of bit detectors form a V shape constellation of bit detectors where the open end of the V shape points in the direction of processing of the broad spiral. Where the two cascades meet one can choose to process a final stripe using either the side information from the cascade having processed the lower portion of stripes, the side information from the cascade having processed the upper portion of stripes, or both side informations.
In addition it is possible to have a bit detector from both cascades process the final stripe. By working both the upper and lower portion of the broad spiral in parallel the processing time is greatly reduced.

An embodiment of the symbol detection method is characterized in that side information is derived from each guard band of the multiple guard bands.

A symbol detector using one of the embodiments of the method according to the invention benefits from a decrease in time required to process the broad spiral or other N-dimensional data.

A playback device using a symbol detector according to the invention benefits from a decrease in time required to process the broad spiral or other N-dimensional data.

A computer program implementing a symbol detector using the methods of the present invention would benefit from a decrease in time required to process the broad spiral of other N-dimensional data.

It should be noted that the channel output is not necessarily sampled on a lattice, nor is it necessary that the channel output are sampled on a similar lattice as the lattice of channel inputs (recorded marks). E.g the channel outputs may be sampled according to a lattice hat is shifted with respect to the lattice of channel inputs (recorded marks). e.g. sampling may take place above edges of the cells of a hexagonal lattice. Also, (signal) dependent oversampling may be applied with higher spatial sampling densities in certain directions as compared to other directions, where these directions need to be aligned with respect to the lattice of signal inputs (recorded marks).

We present a number of ideas which enable to substantially improve the bit-error rate (bER) of the stripe-wise bit-detector for 2D optical storage. A total of <u>nine</u> separate measures have been derived. They are presented in the following paragraphs in order of decreasing priority. The first <u>four</u> measures are the most serious candidate subjects for patent filing: they are briefly described in this abstract.

<u>"Measure 1"</u> is concerned with an extension of the concept of branch metrics to be used for the processing along a Viterbi-trellis of a stripe, involving (i) signal waveform samples of bits outside of the stripe, thus not belonging to the states of the Viterbi processor for the stripe considered and (ii) the introduction of reduced weights smaller than the maximum weight (set equal to 1) for the separate terms in the branch metric that are related to the different bit-rows within the stripe, and (iii) the introduction of cluster-driven weights due to signal-dependent noise characteristics.

<u>"Measure 2"</u> is concerned with a V-shaped progression of stripes in a broad spiral, starting from the two guard bands of the spiral, proceeding towards the center of the broad spiral.

<u>"Measure 3"</u> is concerned with a sequence of stripe-wise iterations of increasing complexity and improving bER performance. First a stripe-wise bit-detector of intermediate complexity and intermediate bER performance with 2-row wide stripes is applied. In a second iteration, another stripe-wise bit-detector of higher complexity and better bER performance with 3-row wide stripes is applied.

<u>"Measure 4"</u> is concerned with a complexity-reduction in the number of states, especially meant for the 3-row wide stripe-wise bit-detector which has a considerable hardware complexity. The orientation of the states on the 2D lattice is arranged such that one particular bit is skipped from the states: in relation to the 2D nature of the intersymbol-interference, the bit that is skipped is the bit that has the largest distance to the branch bit in the output bit-row of the stripe, thus being of lowest significance. The orientation of the states on the bit-lattice depends on the sliding position of the stripe along the 2D broad spiral, and thus on the location of the output bit-row of the stripe.

The detector produces an output row, which is a detected row closest to the predefined data, or most reliable data.

## 1. Aim

*The Context.*

The context of this invention disclosure (ID) is the design of a bit-detection algorithm for information written in a 2D way on a disc or a card. For instance, for a disc, we can still think of a spiral as is the case for 1D recording, albeit a *broad* spiral, that consists of a number of bit-rows that are perfectly aligned one with respect to the other in the radial direction, that is, in the direction orthogonal to the spiral direction. The bits are stacked on a regular quasi close-packed two-dimensional lattice. Possible candidates for a 2D lattice are: the hexagonal lattice, the square lattice, and the staggered rectangular lattice. In this ID, we will focus on the former, since the hexagonal lattice enables the highest recording density. 2D Optical Storage on *a broad spiral* based on an hexagonal lattice is the main carrier of a European Project, called TwoDOS, which has started on April 1st 2002 (see Ref.[1] for the project description). The TwoDOS format is schematically shown in Fig. 1.

*High Storage Densities.*

Further, we aim at such ambitious recording densities that the traditional "eye" is closed. In such a regime, the application of a straightforward threshold detection will lead to an unacceptably high bit error rate ($10^{-2}$ to $10^{-1}$, dependent on the storage density), prior to ECC decoding. Typically, the *symbol* or *byte* error-rate (BER) for random errors in the case of a byte-oriented ECC (like the picket-ECC as used in the Blu-Ray Disc Format, BD) must be not larger than typically $2\ 10^{-3}$; for an uncoded channel bitstream, this corresponds to an upper bound on the allowable *channel-bit* error rate (bER) of $2.5\ 10^{-4}$ (assuming independent bit-error events in a byte).

*PRML-type of Bit-Detection.*

On the other hand, full-fledged PRML type of bit-detectors would require a trellis which is designed for the complete width of the broad spiral, with the drawback of an enormous state-complexity. For instance, if the

horizontal span of the tangential impulse response (along the direction of the broad spiral) is denoted by $M$, and if the broad spiral consists of $N_{row}$ bit-rows, then the number of states for the full-fledged "all-row" Viterbi bit-detector becomes $2^{\wedge}((M-1)\ N_{row})$ (where ^ denotes exponentiation). Each of these states has also $2^{\wedge}(N_{row})$ predecessor states, thus in total the number of *branches* or transitions between states equals $2^{\wedge}(M\ N_{row})$ . The latter number (number of branches in the Viterbi trellis) is a good measure for the hardware complexity of a 2D bit-detector.

*Stripe-Wise Bit-Detection*

Ways to largely circumvent this exponentially growing state-complexity are disclosed in Ref. [2]. In the latter patent application, it is shown how the state-complexity can be reduced by a stripe-based PRML-detector, and iterating from one stripe towards the next (stripes are a set of contiguous "horizontal" bit-rows in the broad spiral). Such a bit-detector is shortly called a *stripe-wise detector*. The recursion between overlapping stripes, the large number of states (already 16 for a stripe of 2 rows, and 64 states for a stripe of 3 rows) and the considerable number of branches (4 for a stripe of 2 rows, and 8 for a stripe of 3 rows), and the recursive character of each individual PRML detector make that the hardware complexity of such a detector can still be quite considerable.

*Purpose of Our Ideas and Trias.*

In this report, we present a number of ideas that are indispensable for a further reduction of the complexity of the stripe-wise bit-detector, meanwhile not sacrificing on its performance.

## 2. HF Signal-Patterns with 2D-Optical Storage on Hexagonal Lattices

We identify the signal-levels for 2D recording on hexagonal lattices by a plot of amplitude values for the complete set of all hexagonal clusters possible. An hexagonal cluster consists of a central bit at the central lattice site, and of 6 nearest neighbour bits at the neighbouring lattice sites. We also make use of the *isotropic* assumption, that is, the channel impulse response is assumed to be circularly symmetric. This implies that, in order to characterize a 7-bit cluster, it only matters to identify the central bit, and the number of "1"-bits (or "0"-bits) among the nearest-neighbour bits (0, 1, ..., 6 out of the 6 neighbours can be a "1"-bit). A "0"-bit is a land-bit in our notation. Typical examples of a 2D "Signal-Pattern" are shown in Figs. 2 and 3: these are computed based on the assumption of scalar diffraction theory.

Note that the isotropic assumption is purely for the purpose of concise presentation. In a practical drive with a tilted disc, the 2D impulse response can have asymmetry. There are two solutions for the latter issue: (i) to apply a 2D equalizing filter restoring a rotationally symmmetric impulse response, and (ii) application of a larger set of reference levels to be used in the branch metric computation, wherein each rotational variant of a given cluster has its own reference level; for this general case, for a 7-bit cluster, consisting of a central bit and its six neighbours, we will have $2^{\wedge}7 = 128$ reference levels, instead of the 14 reference levels in case of the isotropic assumption of above.

The channel bits that are written on the disc are of the land type (bit "0") or of the pit-type (bit "1"). With each bit a physical hexagonal bit-cell is associated, centered around the lattice position of the bit on the 2D hexagonal lattice. The bit-cell for a land-bit is a uniformly flat area at *land*-level; a pit-bit is realized via mastering of a (circular) pit-hole centered in the hexagonal bit-cell. The size of the pit-hole is comparable with or smaller than half the size of the bit-cell. This requirement eliminates the "signal folding" issue, which would arise for a pit-hole that covers the full area of the hexagonal bit-cell (see Ref. [3]): in such case, both for a cluster of all zeroes (all-land) as well as for a cluster of all ones (all-pit), a perfect mirror results, with identical signal levels for both cases. This ambiguity in signal levels must be avoided since it hampers reliable bit-detection. For an intermediate storage density, the signal folding is shown in Fig. 2, with optimized pit-hole sizes on the left, and the largest possible circular pit-holes on the right.

Assuming a broad-spiral consisting of 11 parallel bit-rows, with a guard band of 1 (empty) bit-row between successive broad spirals, the situations shown in Fig. 3 correspond to a storage density increase with a factor of 1.4x and 2.0x, respectively, compared to traditional 1D optical recording of the 3rd generation (as used in e.g. Blu-ray Disc, abbreviated as BD). In each plot, there are two sets of signals drawn: the l.h.s. set corresponds with a central bit equal to "0", the r.h.s. set corresponds with a central bit equal to "1". Note that the amount of signal overlap between the two sets (with different central bit-values) increases with increasing storage density.

## 3. Description of Stripe-Wise Bit-Detector (Prior Art, see Ref. [2])

### 3.1. States, Trellis Structure and Branch Metrics

We first explain the basic structure of the trellis as shown in Fig. 4 (addressing the practical case of a 3-row stripe). The tangential span of the 2D impulse response is assumed to be 3 bits wide, a situation that meets the practical conditions for the high-density TwoDOS Phase-2 situation. A state is specified by two columns

extending over the full radial width of 3 rows of the stripe. There are thus in this example exactly 2^6=64 states. The pace of the Viterbi bit-detector goes with the frequency of emission of a 3-bit column. Emission of a 3-bit column corresponds with a state transition from a so-called *departure* state $\Sigma_m$ (formed by the green and red 3-bit columns) to a so-called *arrival* state $\Sigma_n$ (formed by the red and blue 3-bit columns). For each arrival state, there are exactly 8 possible departure states and thus 8 possible transitions. A transition between two states is called a *branch* in the standard Viterbi/PRML terminology. For each transition, there are thus two states and thus a total of 9 bits that are completely specified by these two states. For each branch, there are a set of reference values which yield the ideal values of the signal waveform at the *branch bits* : these ideal values would apply if the actual 2D bit-stream along the stripe would lead to the considered transition (in the noise-free case). With each transition, we can associate a *branch metric* which gives a kind of "goodness-of-fit" or "figure-of-merit" for the considered branch (or transition) based on the differences that occur between the observed "noisy" signal waveform samples (denoted by *HF*) and the corresponding reference levels (denoted by *RL*). Note that the noise on the observed samples of the waveform can be due to electronic noise, laser noise, media noise, shot noise, residual ISI beyond the considered span of the 2D impulse response etc. It is common practise to consider as the branch bits (at which these differences for the figure-of-merit are to be measured) the bits that are common to both states that constitute the branch: in our example, this would be the 3 bits of the red column at the intersection of the two states. Thus, if $k$ denotes the tangential index at the position of the intersectiong column, and $l$ denotes the top bit-row of the stripe, the branch metric $\beta_{mn}$ between states $\Sigma_m$ and $\Sigma_n$ is given by:

$$\beta_{mn} = \sum_{j=0}^{2} \left| HF_{k,l+j} - RL(\Sigma_m \to \Sigma_n, j, l) \right|^2$$

In the above formula, we have assumed a quadratic error measure for the figure-of-merit ($L_2$ – norm), which is optimum for the assumption of additive white gaussian noise (AWGN). It is also possible to use or error measures, like the absolute value of the difference (known as $L_1$ – norm). For the determination of a reference level for a bit at a given location $k$, $l+j$ on the 2D lattice, we need the values of the six surrounding bits around the location $k$, $l+j$ together with the value of the central bit: these 7 bits uniquely specify the reference level to be used for the considered state transition or branch at the considered bit-location.

### 3.2. Operation of the Stripe-Wise Bit-Detector

We describe now the standard way of operation of the stripe-wise bit-detector as outlined in Ref. [2] (see Fig. 5). A stripe consists of a limited number of bit-rows (e.g. two or three; for Fig. 5, we consider now the practical case of two bit-rows in a stripe. Note that in Fig. 5, a bit-row is bounded by two horizontal lines at its edges.). The number of stripes is equal to the number of bit-rows. A set of Viterbi bit-detectors (denoted as V00, V01, ... in Fig. 5) is devised, one for each stripe. The bits outside of a given stripe that are needed for the computation of the branch metrics, are taken from the output of a neighbouring stripe, or are assumed to be unknown (in a first iteration; the unknown bits may be set to zero). The 1st top-stripe (containing as its top row, the bit-row closest to the guard band) is processed by V00 without any delay at its input; it uses the bits of the guard band as known bits. The output of the 1st stripe are the bit-decisions in the 1st bit-row. The 2nd stripe contains the 2nd and 3rd bit-rows, and is processed by V01 with a delay that matches the back-tracking depth of the Viterbi-detector of the 1st stripe, so that the detected bits from the output of the 1st stripe can be used for the branch metrics of the 2nd stripe. This procedure is continued for all stripes in the broad spiral: the full procedure from top to bottom of the broad spiral is considered to be one iteration of the stripe-wise detector. Subsequently, this procedure can be repeated starting again from the guard band at the top: for the bits in the bit-row just below the bottom of a given stripe, the bit-decisions from the previous iteration can be used. As a reference, we have included an alternative bit-detection scheme as proposed in Ref. [4,5,6] with independent bit-detectors, one for each set of 3 rows (see Fig. 6): each of the 3-row Viterbi bit-detector has as output the bit-decisions on its central bit-rolw. The performance of the latter type of detector at the high capacities for which 2D optical storage is aiming at, is much inferior to that of the stripe-wise bit-detector.

For the sake of completeness, we add that in Fig. 5 with the top-to-bottom processing of successive stripes, the last stripe processor V10 is assumed to output its top bit-row. Another implementation is possible here: we could omit V10, and alter the 2-row stripe processor V09 which comprises the two bottom rows of the spiral such that it outputs both rows simulataneously.

## 4. New Measures for Optimization of Trade-Off between Performance and Complexity of Stripe-Wise Bit-Detector

### 4.1. Figure-of-Merit used as Optimization Criterion in the Stripe-Wise Bit-Detector

#### 4.1.1. Weights of Bits within a Stripe

*Problem*

In the description of the stripe-wise bit-detector, we have explained that a stripe is shifted from the top of the broad spiral in the downward direction towards the bottom of the spiral. The stripe shifts row per row downwards. Each stripe has as its output the bit-decisions of the top bit-row of the stripe. That output bit-row is also used as side-information for the next position of the stripe (shifted one bit-row downwards). The bit-row just across the bottom of the stripe on the other hand still needs to be determined in the current iteration, so we can only use the initialization bit-values for that bit-row in the first iteration of the stripe-wise bit-detector, or in any subsequent iteration, we can use for that bit-row the bit-decisions resulting from the previous iteration of the stripe-wise bit-detector. Therefore, the bit-decisions in the upper bit-row are more reliable than the bit-decisions in the bottom bit-row: this is the reason why the output of one stripe is its top bit-row. Also, for the computation of the required reference levels in the bottom bit-row, we need as explained in Section 3, the six nearest neighbour bits of the branch bit in the bottom bit-row; two of these nearest neighbour bits are located in the bit-row just below the stripe considered, and only preliminary bit-decisions (from the previous iteration) are available for these bits. Consequently, in case of bit-errors in these two bits in the bit-row below the current stripe, these errors will affect the selected branches in the surviving path along the Viterbi trellis: actually, the bit-errors in these two bits may be compensated by selecting the wrong bits in the states along the stripe, so that the error measure at the bottom branch bit can be kept low enough. Unfortunately, this balancing will propagate errors towards the top bit-row of the stripe, which should be prohibited.

*Solution*

A solution to the above drawback is to reduce the relative weight for the bottom branch bit in the figure-of-merit from the full 100% to a lower fraction. With $w_i$ denoting the weight of the branch bit in the $i$-th row of the stripe, the branch metric becomes:

$$\beta_{mn}=\sum_{j=0}^{2} w_j \left|HF_{k,l+j}-RL\left(\Sigma_m \to \Sigma_n, j, l\right)\right|^2$$

By choosing the weight of the bottom row in the stripe to be much lower than 1, the negative influence of the unknown or only preliminary known bits in the bit-row just below the current stripe is largely reduced. This is shown schematically in Fig. 7. The weights of the respective contributions of the signal waveforms to the branch metrics can also be varied from one iteration to the next (because the bit-decisions at the surrounding bits become gradually more and more reliable).

For the sake of completeness, note that the above description applies to a top-to-bottom processing of the stripes, wherein the output of each stripe is its top bit-row, and the weight of the bottom bit-row is reduced. However, for the opposite processing order, from bottom-to-top, the output of each stripe is its bottom bit-row, and the weight of the top bit-row is reduced.

#### 4.1.2. Extra Signal-Waveform Samples in Branch Metrics at Bits in Bit-Rows that are exterior to the Stripe

*Problem: Leaked-Away Information in 2D Bit-Detection*

For high-density 2D optical storage (factor 2x capacity of BD), the 2D impulse response of the (linearized) channel can be approximated to a reasonable level of accuracy by a central tap with tap-value $c_0$ equal to 2, and with 6 nearest-neighbour taps with tap-value $c_1$ equal to 1. The total energy of this 7-tap response equals 10, with an energy of 6 along the tangential direction (central tap and two neighbour taps), and an energy of 2 along each of the neighbouring bit-rows (each with two neighbour taps). This is shown schematically in Fig. 8.

From these energy considerations, one of the main advantages of 2D modulation can be argued to be the aspect of "joint 2D bit-detection", where all the energy associated with each single bit is used for bit-detection. This in contrast to 1D detection with standard cross-talk cancellation, where only the energy "along-track" is being used, thus yielding a 40% loss of energy per bit.

A similar argumentation holds when we consider bit detection at the edges of a 2D stripe (for which we want to output the top bit-row). Of the order of 20% of the signal-energy of the bits in the top-row has leaked away in the samples of the signal waveform of the two samples in the bit-row just above the stripe: these two samples are located at nearest neighbour sites of the bit in the top row of the current stripe. The other 20% leaking away from the top bit-row is leaking away in the bit-row below the top bit-row in the stripe: this energy is used because the stripe (of at least two bit-rows wide) comprises also the bit-row below the top bit-row of the stripe. Consequently, not using the leaked away information, that has been leaking away in the "upward" direction

(when the top bit-row is the output of the considered stripe), would lead to a loss in bit-detection performance at the top row of the stripe.

*Solution*

The solution to the above drawback is to include the HF-samples in the bit-row above the stripe in the computation of the figure-of-merit. Note that only the samples of the signal waveform of that row do matter here, and that the bits in that row are *not* varied since they do not belong to the set of bits that are varied along the trellis and states of the Viterbi-detector for the stripe considered. Denoting the row-index of the bit-row above the stripe by $l$-$1$, the branch metric is denoted by (with the running index $j$ now starting from "-1"):

$$\beta_{mn} = \sum_{j=-1}^{2} w_j \left| HF_{k,l+j} - RL(\Sigma_m \rightarrow \Sigma_n, j, l) \right|^2$$

This extension of the computation of the branch metrics with inclusion of the row of signal samples in the bit-row above the stripe is schematically drawn in Fig. 9. Note that in the computation of the reference levels, all the required bits within the stripe are set by the two states that consitute a given branch, and all the required bits outside the stripe are determined by the previous stripe in the current iteration of the stripe-wise bit-detector, or by the previous iteration of the stripe-wise bit-detector.

For the sake of completeness, note that the above description applies to a top-to-bottom processing of the stripes, wherein the output of each stripe is its top bit-row, and the extra bit-row that is accounted for in the branch metrics, is the row just above the stripe, with index $j$=$-1$. However, for the opposite processing order, from bottom-to-top, the output of each stripe is its bottom bit-row, and the extra bit-row that is accounted for in the branch metrics, is the row just below the stripe, with index $j$=$3$ (for a 3-row stripe).

#### 4.1.3. Normalization by a Cluster Dependent Noise Variance

In detection theory, it is a well-known known fact that in an optimal Viterbi detector, the branch metrics are (negative) log-likelihoods of the channel input bits given the observed channel output values. Already in Section 3.1 it was argued that the branch metric formula

$$\beta_{mn} = \sum_{j=0}^{2} \left| HF_{k,l+j} - RL(\Sigma_m \rightarrow \Sigma_n, j, l) \right|^2$$

derives its validity from the assumption that the noise is Additive, Gaussian and White. The squares inside the sum above stem from the logarithm of the Gaussian probability density function of the noise $g_{mn}$ which also contains a square,

$$-\log(\Pr\{g_{mn} = g\}) = \frac{1}{2}\log(2\pi N) + \frac{g^2}{2N}$$

The whiteness assumption of the noise implies that different noise components are statistically independent, so that their probability density functions can be multiplied. Therefore, their log-likelihood functions can be added, as in the $\beta_{mn}$ formula.

The problem we want to consider here, is that e.g. for an optical recording the variance of the noise $N$ may depend on the central input bit of a given channel output $HF_{k,l+j}$ and its cluster of nearest neighbour inputs. For example, in case laser noise is dominant, larger channel outputs $HF_{k,l+j}$ carry more (multiplicative) laser noise (which is usually referred to as 'RIN', "relative intensity noise"). This leads to the question what value of the noise $N$ to use in the branch metric formula for $\beta_{mn}$ ?

The solution to this problem is very simple. Based on a table of the cluster-dependent noise variances, we make a table for the noise variance $N(\Sigma_m \rightarrow \Sigma_n, j)$ as a function of the state transition $(\Sigma_m \rightarrow \Sigma_n)$ and the row index $j$, and we divide by the adjusted value of $N$ in the branch metric formula,

$$\beta_{mn}=\sum_{j=0}^{2} w_j \frac{\left|HF_{k,l+j}-RL(\Sigma_m \rightarrow \Sigma_n, j, l)\right|^2}{N(\Sigma_m \rightarrow \Sigma_n, j, l)}$$

When the noise is really dependent on the cluster and on the central input bit of a given channel output, taking account of this as in the branch metric formula above makes the branch metrics more closely equal to the log-likelihood functions as stated in the introduction of this subsection. This in general results in an improvement of the resulting bit error rate at the bit-detector output.

### 4.2. Two-sided V-shaped Progression of Stripes in a Broad Spiral e.g. starting from the Guard Bands towards the Center of the Broad Spiral

One iteration of the stripe-wise bit-detector may consist as described above out of a successive processing of stripes starting from the guard band on top of the broad spiral towards the guard band at the bottom of the broad spiral. Instead, one can start with stripes from both guard bands and successively process a number of stripes proceeding from both sides towards the middle of the broad spiral. Successive stripes are arranged in a V-shape as can be seen in Fig. 10 for the practical case of a 11-row broad spiral and stripes consisting of two bit-rows. The Viterbi-detectors "V00", "V02", ..., "V08" are cascaded one after the other with mutual delay to allow for back-tracking of the respective detectors, and the cascade starts from the top guard-band towards the center of the broad spiral; each of these Viterbi-detectors has as output the bit-decisions for the top bit-row. Each of these Viterbi-detectors also uses the signal waveform samples at the bit-row above the stripe as additional extra row in the branch metrics; the weight of the signal waveform samples in the bottom row of the stripe is reduced below the maximum value (set equal to 1). In analogy, the Viterbi-detectors "V01", "V03", ..., "V07" are cascaded one after the other (also with mutual delay for back-tracking purposes) starting from the bottom guard-band towards the center of the broad spiral; each of these detectors has as output the bit-decisions for the bottom bit-row. Each of these Viterbi-detectors also uses the signal waveform samples at the bit-row below the stripe as additional extra row in the branch metrics; the weight of the signal waveform samples in the top row of the stripe is reduced below the maximum value (set equal to 1). These two sets of cascaded Viterbi-detectors have a mutual mirror-type of relationship. Finally, the two cascades of stripes are terminated in the middle of the broad spiral with a last stripe "V09", which is the only stripe that has as output its two bit-rows, and which has extra exterior bit-rows on both sides of the stripe (of which the signal waveforms are included in the computation of the branch metrics of that stripe); also the weights of all signal waveforms at the branch-bits are set equal to the maximum value 1 (since the bit-rows at both sides of this stripe have been determined during execution of the two cascades of Viterbi-detectors in all previous stripes).

With the V-shaped stripe-wise bit-detector, the propagation direction of "bit-reliability" is from the known bits of the guard band towards the bit-row in the middle of the broad spiral, which are thus the largest distance from the guard bands: the "known" information is propagated from both sides towards the middle, which is a better approach than propagating from top to bottom of the broad spiral.

The idea of this subsection can be generalized in the following way: the stripes can be cascaded as two sets forming a V-shaped configuration between *any* pair of two bit-rows in the 2D area that have a significantly higher bit-reliability, so that they can serve as anchor points from which successive stripes can propagate in a two-sided way towards each other in the middle area between the two rows with high bit-reliability. In the particular case (treated above) of a broad spiral with two guard bands with bits that are known to the detector, the bit-reliability of the two anchor bit-rows is 100%. Another example is the case of a 2D format with an extra bit-row in the middle of the spiral, that is encoded such that it has a higher bit-reliability than the other rows; then, two V-shaped progressions of stripes can be devised, one operating between the center bit-row and the upper guard band, the other operating between the same center bit-row and the lower guard band (see Fig. 11). For instance, the center bit-row may be channel encoded with a 1D runlength limited (RLL) channel code that enables robust transmission over the channel: for instance, a d=1 RLL channel code removes some of the clusters (those with a "1" central bit and all six "0"'s as neighbour bits, and vice versa) in the overlap area of the signal patterns, hereby increasing the robustness of bit-detection on the one hand, but reducing the storage capacity for that row on the other hand because of the constrained channel coding.

During back-tracking of a Viterbi-processor for a given stripe, it is an option to output all bit-rows of the stripe so that a bit-array with the most recent bit-estimates are stored. The purpose of this measure is to achieve a more

uniform architecture for the Viterbi-processors in the top-half, bottom-half and central area of a V-shaped bit-detection scheme.

### 4.3. Optimized Complexity in terms of Number of Bit-Rows for Subsequent Iterations of the Stripe-Wise Bit Detector

When limiting the number of iterations of the stripe-wise bit-detector to only two, the best performance in terms of bit-error rate (bER) is achieved when the last iteration is the most powerfull one, going down as much as possible in bER: therefore, this last iteration must be subject to the smallest error floor that is achievable. That last iteration needs at its input the output of the previous (first) iteration, which needs to be of high enough quality. It is observed from simulation experiments that when we use 3-row stripes for the second iteration, it is satisfactory to use 2-row stripes for the first iteration. Fig. 12 shows a succession of two V-shaped iterations, the first iteration on the right-hand side comprising 2-row stripes, the second iteration on the left-hand side comprising 3-row stripes. The explanation of the different Viterbi-detectors has been given for the 2-row stripes in the previous paragraph. For completeness, we give the detailed explanation of the various Viterbi-detectors for the 3-row stripes as used in Fig. 12. The 3-row Viterbi-detectors U00, U02, ..., U06 are cascaded one after the other starting from the guard band at the top of the broad spiral, and have as output the top bit-row of each stripe; the weight in the branch metrics of the signal waveform samples in the bottom row are reduced below 1; the branch metrics are extended to include the signal waveform samples of the bit-row just above the stripe. In analogy, the 3-row Viterbi-detectors U01, U03, ..., U07 are cascaded one after the other starting from the guard band at the bottom of the broad spiral, and have as output the bottom bit-row of each stripe; the weight in the branch metrics of the signal waveform samples in the top row are reduced below 1; the branch metrics are extended to include the signal waveform samples of the bit-row just below the stripe. These two sets of cascaded Viterbi-detectors have a mutual mirror-type of relationship. Finally, the two cascades of 3-row stripes are terminated in the middle of the broad spiral with a last stripe "U08", which is the only stripe that has as output its three bit-rows, and which has extra exterior bit-rows on both sides of the stripe (of which the signal waveforms are included in the computation of the branch metrics of that stripe); also the weights of all signal waveforms at the branch-bits are set equal to 1 (since the bit-rows at both sides of this stripe have been determined during execution of the two cascades of Viterbi-detectors in all previous stripes).

Note that the hardware complexity (which is conveniently measured in terms of the number of states times branches in a Viterbi-detector) is a factor 8x larger for a 3-stripe Viterbi than it is for a 2-stripe Viterbi. So it is advantageous to devise additional measures that may reduce the hardware complexity of the 3-stripe Viterbi, without sacrificing its performance too much: this is the topic of the next section.

### 4.4. Tangential Local Sequence Feedback for Reduction of State-Space Complexity

Fig. 13 shows the description of the states for the Viterbi-detector for a 3-row stripe. Note that the orientation of the bit-columns in the states is of no relevance for their performance. We have redrawn the orientation for the stripe that is sliding in the downward direction, from the top to the center of the broad spiral, since it is required for the additional measures that we are going to describe in the sequel of this subsection: this re-orientation of the states on the 2D lattice is an essential feature of the present invention.

*Single-Bit Local Sequence Feedback*

The number of states (and consequently, the hardware complexity of the Viterbi) can be reduced by a factor of two by considering 5-bit states instead of 6-bit states as shown in Fig. 14. The bits on the states are arranged such that the "missing" bit (indicated by the "X" in Fig. 14) has the largest distance relative to the branch bit (at the intersection of the two states) that is located in the output bit-row of the considered stripe: in that sense, the "missing" bit is the bit of which the degradation of bER performance caused by its omission is minimal. The latter degradation is forced to be as minimal as possible due to the above re-orientation of the states on the 2D lattice. Note that we still compute branch metrics in the branch bits which form a column extending over the full width of the stripe (which is not identical to bits at the intersection of two states, the latter consists of only two bits) possibly extended with samples from the adjacent bit-row at the side of the output bit-row of the stripe (see Section 4.1.2). The "missing" bit is only needed for the computation of the contribution to the branch metric from the branch bit that is located at largest distance from the output bit-row of the stripe; recall that, for each branch bit, we need to know its 6 nearest neighbour bits to derive the required reference level. In a state-

transition, there are two bits in common that belong to each of both states; the number of branches is indeed 8, related to the three (5-2=3) bits that are not common to both states.

The missing bit is determined through a procedure known as "local sequence feedback" (see Ref. [7] for the case of 1D signal processing). The procedure works as follows. For a given branch between two states, each of 5 bits, we know already a total of 8 bits directly from the states. At this stage of Viterbi-processing, for each of the possible alternatives for the departure state of a given branch, we already know the best "predecessor" state for the considered departure state of the given branch; that best predecessor state specifies the bit-value on the missing-bit position, so that we have the original total of 9 bits that are required to compute all contributions to the branch metrics (in addition, we have at our disposition of course also previous and/or preliminary bit-decisions on the neighbour bit-rows outside the bit-rows of the considered stripe).

*Two-Bit Local Sequence Feedback*
Analoguously, the number of states can be reduced by a factor of four by considering states consisting of only 4 bits, as is shown in Fig. 14bis. Note that there is only one possible location for the second missing bit when we have located the first missing bit in the most remote corner of the set of 9-bits, as described above: the second missing bit must be in the first column on the left-hand side of the set of 9 bits, and it must be at the largest possible distance from the branch bit (the center bit) in the output bit-row in order to make the sequence feedback as efficient as possible. Upon a state-transition, there is only one bit in common that belongs to each of both states; the number of branches-per-state is indeed 8, related to the three (4-1=3) bits that are not common to both states. The two "missing" bits are indicated by "X" in Fig. 14bis. The two missing bits are determined in a similar way through "local sequence feedback" as described above for the case of a single missing bit. For each of the possible alternatives for the departure state of a state-transition or branch, we have determined its best predecessor state: that best predecessor state specifies the bit-values of the two "missing" bits.

Similar measures as described in this section can be devised for stripes of different (radial) widths, e.g. for the case of a 2-row stripe with a "local sequence feedback on a single-bit" as is shown in Fig. 14tris.

*Three-Bit Local Sequence Feedback (only for 3-row stripe)*
Analoguously, the number of states can be further reduced by a total factor of eight by considering states consisting of only 3 bits (instead of 6 bits), as is shown in Fig. 14''''. Upon a state-transition, there is no bit in common that belongs to each of both states; the number of branches-per-(arrival)state is indeed 8, related to the three bits of the departure state. The three "missing" bits are indicated by "X" in Fig. 14''''. The three missing bits are determined in a similar way through "local sequence feedback" as described above for the case of a single missing bit or of two missing bits. For each of the possible alternatives for the departure state of a state-transition or branch, we have determined its best predecessor state; that best predecessor state specifies the bit-values of the three "missing" bits.

**4.5. Initialization Step with Threshold Detection with row-dependent Adapted Threshold Levels**

Prior to any Viterbi bit-detection, it is advantageous to have some preliminary bit-decisions albeit at a relatively poor bit-error rate (bER) performance. For instance, at one side of each stripe, we have bits that have been determined from the previous stripe or that are set to zero when the stripe is located directly next to the guard band; at the other side of the stripe, we need bit-decisions in order to be able to derive reference levels for the bits in the neighbouring bit stripe within the stripe: these bit-decisions can be derived from a previous iteration of the stripe-wise bit-detector, or from preliminary bit-decisions when the 1st iteration of the stripe-wise bit-detector is being executed. These preliminary decisions can just be obtained by putting all bits to zero, which is not such a clever idea.
A better approach is to apply threshold detection based on threshold levels (or slicer levels) that depend on whether the row is neighbouring the guard band (consisting of all zeroes) or not, as shown in Fig. 15 (for the case of 2D optical storage at a capacity of 2x that of Blu-ray Disc, BD). In the case of a bit-row neighbouring the guard band, some cluster-levels are forbidden as indicated by the X'es in Fig. 15: therefore, the threshold level is shifted upwards: it is computed as the level between the cluster-level for a central bit equal to 0 and three 1-bits as neighbour, and the cluster-level for a central bit equal to 1 and one 1-bit as neighbour. The expected bit-error rate of this simple threshold detection is then, for this case, equal to 2/32, which is about 6%. In the case of a bit-row that is *not* neighbouring the guard-band, the threshold level is computed as the level between the cluster-level for a central bit equal to 0 and four 1-bits as neighbour, and the cluster-level for a central bit equal to 1 and two 1-bits as neighbour. The expected bit-error rate of this simple threshold detection is then, for this case, equal to 14/128, which is about 11%. Although these bERs are quite high, they are considerably better (especially at the bit-rows neighbouring the guard bands) than the 50% bER obtained

through coin tossing. These preliminary bit-decisions obtained prior to the execution of the stripe-wise bit-detector can also be used as input for the adaptive control loops of the digital receiver (e.g. for timing recovery, gain- and offset-control, adaptive equalization etc.) Note that the above derivation of the proper slicer levels depends on the actual 2D storage density chosen and the resulting overlap of signal levels in the "Signal Patterns". .

**4.6. Idea to, During Later Iterations when the Bit Error Rate is Small, Piecewise Skip the Processing of the Stripewise Bit-Detector**

The preferred embodiment of the stripewise bit-detector for a storage density of 2.0xBD consists of two iterations:

- A first iteration with stripes of two rows.
- A second iteration with stripes of three rows.

See Fig. 12. Simulations have shown that after the first iteration, the bit error rate is around 0.002. The purpose of the second, more complex iteration is to deal with the more complex error patterns for which larger groups of neighbouring bits are in error. Observe that a fraction of 0.998 of the input bits of such a second iteration are already correct. As in sequential decoding theory, the presence or absence of bit errors in a given piece of a stripe can be detected from the presence or absence of an increase of the cumulated branch metrics (i.e. path metrics) along the survivor path after the first iteration. The idea presented here is to apply the computations intensive 3 row bit-detector only over pieces (adjacent columns) of a stripe where such increases of path metrics are detected one iteration earlier, when these bits are output. If we process the fraction 0.002 of erroneous bits plus adjacent columns to the left and right of these erroneous spots, we end up processing e.g. only a fraction of 0.1 of a stripe. Thus, the computational complexity of the second iteration – which is the most computational intensive iteration – can be lowered by a factor of 10.

**4.7. Idea to Use What is Known as "Reduced State Techniques" (see Ref. [8]) for the Last Iteration**

In this section we present an alternative to the technique of the previous Section 4.6, to take advantage of the fact that the last, computationally most intensive iteration is presented with a broad spiral of input bits that already has a very low bit error rate (e.g. 0.002 bER).

The last iteration can work with bits that are defined differential with respect to the output bits of the previous iteration. This way, if the output bits of the previous iteration are already correct, the last iteration will find the all zeroes broad spiral as "best differential bit values." Due to the low input bit error rate, the assumption of error free inputs, is only an approximation. When we do not apply sequence feedback as in Section 4.4, the product of the number of states and number of branches in the last iteration with 3-row stripes is $2^9$=512. In practise, it turns out that the number of (branch,state) pairs that actually need to be taken into account in the last iteration can be reduced to less than half that number, e.g. 200. The (branch,state) pairs that do occur, correspond to the error patterns in the data at the input of the last iteration. They are specific patterns, not all combinations occur. E.g. (branch,state) pairs that lead to a 3x3 plane of bits of Hamming weight larger than e.g. 5 are rare, and can be neglected (see schematic representation in Fig. 16). In this way, the computational complexity of the last iteration can be reduced. A disadvantage of this technique is that is makes a hardware design less regular. Yet another disadvantage of this technique is that the "skipping of state-transitions" is tailored to the error-patterns at hand in the practical receiver and optical drive (for 2D optical storage): if the capacity changes slightly, and/or the experimental characteristics of the optical read-out of the disc are altered, the "skipping of state-transitions" might need to be changed, requiring new hardware, making this technique less attractive; on the other hand, the full-fledged 3-row Viterbi for each stripe of 3 rows is more generic and works "under all seasons".

**4.8. Local Sequence Feedback in the Radial Direction for Initialization Step**

In Section 4.5, we described a way (via threshold detection) to initialize the bits to preliminary bit-values as they are needed in the neighbouring bit-row on one side of the stripe (in the sliding direction of the stripe). The threshold detection of Section 4.5 was "stand-alone", not needing any bits around the bit-position where the bit-decision has to be taken. Another way to produce preliminary decisions is coupled to the states of the Viterbi

used along a given stripe, as is shown in Fig. 17 for bits "a" and "b" in the bit-row just below the stripe. The preliminary bit-decisions on these bits are made conditionally dependent on the transition between two states of the Viterbi trellis of the stripe. For such a transition, all bits in the bottom row of the stripe are specified: they condition two of the six neighbours of bit "a" (by the green bit and the red bit at the bottom of the l.h.s. state $\Sigma_m$), and also two of the six neighbours of bit "b" (by the red bit and dark-blue bit at the bottom of the r.h.s. state $\Sigma_n$). If these state-bits are equal to 1 and 0 or 0 and 1, we use as threshold level just the level as shown in Fig. 15 on the l.h.s. (the outer cluster-levels either with all six 0-bits or all six 1-bits as neighbours in the signal-pattern are not possible on condition of the considered state-transition, but the threshold level remains the same). On the other hand, if these state-bits are equal to 0 and 0, we have the situation that resembles the case of a bit-row next to the guard band, with an adapted threshold level as on the r.h.s. of Fig. 15 (the four bottom cluster-levels in the signal-pattern are not possible to occur on condition of the considered state-transition). Similarly (but not shown explicitly), if these state bits both are equal to 1, the four top cluster-levels in the signal pattern cannot occur, and a threshold level is derived between the cluster level for the central bit equal to 0 with five 1-bits as neighbours, and the cluster-level for the central bit equal to 1 with three 1-bits as neighbours.
Note that the actual setting of the threshold levels depends on the targetted storage density: the above results were derived for the case of a capacity equal to 2x BD. Further note that not a single preliminary bit-decision can be obtained for a given bit-position in this way, but a bit-decision that depends on the state-transition being processed in the Viterbi-trellis of the considered stripe.

### 4.9. Operation of Stripe-Wise Bit-Detection along Stripes oriented in other Directions than the tangential Direction along the Axis of the Broad Spiral

In Fig. 18, two diagonal orientations of the stripe on the 2D hexagonal-lattice are shown. For these diagonal orientations, the shifting of the stripe takes place along the direction of the broad spiral: this implies that the Viterbi processing with state-termination at the guard bands where the bits are known to be zero, has to be completed before the shifting over the distance of one bit along the tangential direction can take place. The latter aspect is a real disadvantage with respect to parallellization of the hardware implementation. Different executions of the stripe-wise bit-detector, operating along different directions, can be cascaded one after the other. Also, more oblique orientations than the ones shown in Fig. 18 can be devised (the ones shown in Fig. 18 are oriented along the basic axes of the 2D hexagonal lattice, with angles of exactly 60 degrees between them).

## 5. Suggestions for Claims (in view of a "Dumped Patent Application")

### 5.1. Coupled to "Measure 1", Section 4.1.

A bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which the branch metrics, which reflect a sum of squared differences or absolute values of differences or any other applicable norm on a set of differences, said difference being computed between a received or observed sample of the signal waveform and a properly determined noise-free reference level that is typical for the branch considered, said branch metrics apply for each of the possible state-transitions along the associated trellis of the Viterbi processing, said branch metrics are generalized with respect to the following aspects:
- each stripe processes a number of bit-rows simultaneously, but has only as output the bit-row at one of its boundaries. The branch metric computation is extended to include the signal waveform samples from the bits in the neighbouring bit-row just exterior to the stripe, and at the side of the output bit-row of the stripe, since the signal energy of the output bit-row has leaked away partly into the samples of said exterior bit-row. The bits in said exterior bit-row beyond the stripe, at the side of the output bit-row, are not varied according to the trellis of the Viterbi-detector, but are determined from a previous position of the stripe, when said exterior bit-row was the output bit-row of said previous position of the stripe.
- the branch metrics are a sum of separate terms, one term for each branch bit considered to contribute to the branch metrics; each term may have a local weight that depends on the position of said branch metric relative to the edges of said stripe, for instance, the weights for branch bits that are far away from the output bit-row at one side of the stripe, may be set to low values;
- each term in the branch metric may be weighted by a transition-dependent and cluster-dependent noise variance, said weighing combatting the influence of signal-dependent noise.

A bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which the branch metrics, which reflect a sum of squared differences or absolute values of differences or any other applicable norm on a set of differences,

said difference being computed between a received or observed sample of the signal waveform and a properly determined noise-free reference level that is typical for the branch considered, said branch metrics apply for each of the possible state-transitions along the associated trellis of the Viterbi processing, said branch metrics are generalized with respect to the following aspects:
- each stripe processes a number of bit-rows simultaneously, but has only as output the bit-row at one of its boundaries. The branch metric computation is extended to include the signal waveform samples from the bits in the neighbouring bit-row just exterior to the stripe, and at the side of the output bit-row of the stripe, since the signal energy of the output bit-row has leaked away partly into the samples of said exterior bit-row. The bits in said exterior bit-row beyond the stripe, at the side of the output bit-row, are not varied according to the trellis of the Viterbi-detector, but are determined from a previous position of the stripe, when said exterior bit-row was the output bit-row of said previous position of the stripe.
- the branch metrics are a sum of separate terms, one term for each branch bit considered to contribute to the branch metrics; each term may have a local weight that depends on the position of said branch metric relative to the edges of said stripe, for instance, the weights for branch bits that are far away from the output bit-row at one side of the stripe, may be set to low values;
- each term in the branch metric may be weighted by a transition-dependent and cluster-dependent noise variance, said weighing combatting the influence of signal-dependent noise where the weight in the branch metric of the bit-row that is exterior to said stripe, is put to zero.

A bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which the branch metrics, which reflect a sum of squared differences or absolute values of differences or any other applicable norm on a set of differences, said difference being computed between a received or observed sample of the signal waveform and a properly determined noise-free reference level that is typical for the branch considered, said branch metrics apply for each of the possible state-transitions along the associated trellis of the Viterbi processing, said branch metrics are generalized with respect to the following aspects:
- each stripe processes a number of bit-rows simultaneously, but has only as output the bit-row at one of its boundaries. The branch metric computation is extended to include the signal waveform samples from the bits in the neighbouring bit-row just exterior to the stripe, and at the side of the output bit-row of the stripe, since the signal energy of the output bit-row has leaked away partly into the samples of said exterior bit-row. The bits in said exterior bit-row beyond the stripe, at the side of the output bit-row, are not varied according to the trellis of the Viterbi-detector, but are determined from a previous position of the stripe, when said exterior bit-row was the output bit-row of said previous position of the stripe.
- the branch metrics are a sum of separate terms, one term for each branch bit considered to contribute to the branch metrics; each term may have a local weight that depends on the position of said branch metric relative to the edges of said stripe, for instance, the weights for branch bits that are far away from the output bit-row at one side of the stripe, may be set to low values;
- each term in the branch metric may be weighted by a transition-dependent and cluster-dependent noise variance, said weighing combatting the influence of signal-dependent noise where the weights in the branch metric of all bit-rows within said stripe, are put equal to each other.

A bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which the branch metrics, which reflect a sum of squared differences or absolute values of differences or any other applicable norm on a set of differences, said difference being computed between a received or observed sample of the signal waveform and a properly determined noise-free reference level that is typical for the branch considered, said branch metrics apply for each of the possible state-transitions along the associated trellis of the Viterbi processing, said branch metrics are generalized with respect to the following aspects:
- each stripe processes a number of bit-rows simultaneously, but has only as output the bit-row at one of its boundaries. The branch metric computation is extended to include the signal waveform samples from the bits in the neighbouring bit-row just exterior to the stripe, and at the side of the output bit-row of the stripe, since the signal energy of the output bit-row has leaked away partly into the samples of said exterior bit-row. The bits in said exterior bit-row beyond the stripe, at the side of the output bit-row, are not varied according to the trellis of the Viterbi-detector, but are determined from a previous position of the stripe, when said exterior bit-row was the output bit-row of said previous position of the stripe.
- the branch metrics are a sum of separate terms, one term for each branch bit considered to contribute to the branch metrics; each term may have a local weight that depends on the position of said branch metric relative to the edges of said stripe, for instance, the weights for branch bits that are far away from the output bit-row at one side of the stripe, may be set to low values;

- each term in the branch metric may be weighted by a transition-dependent and cluster-dependent noise variance, said weighing combatting the influence of signal-dependent noise where the weights are iteration-dependent.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where the bit-rows with high bit-reliability are the guard bands of a broad spiral that contain bits that are a-priori known to the bit-detector.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where the bit-rows with high bit-reliability are the guard bands of a broad spiral that contain bits that are a-priori known to the bit-detector, where the bits in the guard band are all set to the same binary bit-value.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where one of the bit-rows with high bit-reliability is a bit-row that is part of a band of bit-rows that has been additionally channel coded to have good transmission properties over the channel.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where one of the bit-rows with high bit-reliability is a bit-row that is part of a band of bit-rows that has been additionally channel coded to have good transmission properties over the channel, where said band of bit-rows comprises exactly one bit-row.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where one of the bit-rows with high bit-reliability is a bit-row that is part of a band of bit-rows that has been additionally channel coded to have good transmission properties over the channel, where said band of bit-rows comprises exactly one bit-row, where said bit-row with high bit-reliability is channel encoded with a runlength-limited modulation code.

a bit-detection method for bit-detection on a 2D array of bits, arranged on a regular 2D lattice, preferably an hexagonal bit-lattice, that is based on a stripe-wise bit-detector, in which stripes are successively processed in a cascaded fashion, starting from the bit-rows in the 2D array of bits that have a considerable higher certainty of bit-reliability, towards the center of the 2D area that is bounded by said two bit-rows of higher bit-reliability, where one of the bit-rows with high bit-reliability is a bit-row that is part of a band of bit-rows that has been additionally channel coded to have good transmission properties over the channel, where said band of bit-rows comprises exactly one bit-row, where said bit-row with high bit-reliability is channel encoded with a runlength-limited modulation code, where said runlength-limited modulation code staisfies the $d=1$ runlength constraint.

CLAIMS:

1 A stripe wise iterative symbol detection method for detecting symbol values of a data block recorded along an N-dimensional channel tube, N being at least 2, on a record carrier of a set of symbol rows, one dimensionally evolving along a first direction and being aligned with each other along at least a second of N-1 other directions, said first direction together with said N-1 other direction constituting an N-dimensional lattice of symbol positions, wherein a stripe is a subset of a row and at least one neighboring row, the iteration of said stripe wise iterative symbol detection comprises:

- estimating the symbol values in a first stripe, using a search based algorithm, a side information derived from a row adjacent to the first stripe, the side information being used in the estimation of said symbol values,
characterized in that a weighing of a contribution of the side information is assigned based on a reliability of the side information

2 A symbol detection method as claimed in claim 1, characterized in that the contribution is a contribution to an objective function of the search based algorithm.

3 A symbol detection method as claimed in claim 2,
characterized in that the search based algorithm comprises the use of contributions internal to the stripe and that the use of the internal contributions comprises assigning an individual weighing of the internal contributions

4 A symbol detection method as claimed in claim 1, 2 or 3,
characterized in that the search based algorithm is a Viterbi algorithm, a sequential decoding algorithm such as a stack algorithm or a Fano algorithm, or a soft-decision output algorithm such as a (Max)(Log)MAP algorithm.

5 A symbol detection method as claimed in claim 4,
characterized in that the side information is an estimated channel input symbol

6 A symbol detection method as claimed in claim 4,
characterized in that the side information is likelihood information about a channel input symbol

7 A symbol detection method as claimed in claim 5 or 6,
characterized in that a further side information, derived from the row adjacent to the first stripe, is being used in the estimation of said symbol values.

8 A symbol detection method as claimed in claim 7,
characterized in that the further side information comprises channel output values

9 A symbol detection method as claimed in claim 8,
characterized in that the channel output values are filtered channel output values

10 A stripe wise iterative symbol detection method as claimed in claim 1, 2, 3, 4, 5, 6, 7, 8 or 9
characterized in that a weighing of the contribution of the side information is highest for side information derived from a symbol detection with a highest reliability.

12 A stripe wise iterative symbol detection method as claimed in claim 11, characterized in that the symbol detection with the highest reliability is a symbol detection from a previous iteration.

12 A stripe wise iterative symbol detection method as claimed in claim 10 or 11, characterized in that the weighing is based on a distance between a position of a symbol value to be detected and a position of a side information symbol position

13 A stripe wise iterative symbol detection method as claimed in claim 12, characterized in that the distance is a distance to a most reliable side information position

14 A stripe wise iterative symbol detection method as claimed in claim 10, 11, 12,or 13, characterized in that the weighing of the contribution of the side information is different for the second detector compared to the first detector

15 A stripe wise iterative symbol detection method as claimed in claim 10, 11, 12, 13 or 14, characterized in that the weighing of the contribution of the side information is different for a second iteration compared to a first iteration.

16 A stripe wise iterative symbol detection method as claimed in claim 15, characterized in that the weighing of the contribution of the side information is higher for the second iteration compared to the first iteration.

17 A symbol detection method as claimed in claim 10, 11, 12, 13 or 14, characterized in that the side information is obtained from a row comprising data which is highly protected using redundant coding.

18 A symbol detection method as claimed in claim 10, 11, 12, 13, or 14, characterized in that the side information is obtained from a row comprising predefined data.

19 A symbol detection method as claimed in claim 17, characterized in that the row comprising data which is highly protected using redundant coding is a guard band

20 A symbol detection method as claimed in claim 17, characterized in that the row comprising data which is highly protected using redundant coding is located centrally between the rows forming the set of symbol rows.

21 A symbol detection method as claimed in claim 19, characterized in that the N-Dimensional channel tube is delimited by one or more guard bands.

22 A symbol detection method as claimed in claim 19, characterized in that side information is derived from each guard band of the one or more guard bands

23 A symbol detector using one of the methods of the claims 1 to 22

24 A playback device comprising a symbol detector as claimed in claim 23

25 A computer program using one of the methods of the claims 1 to 22

PHILIPS

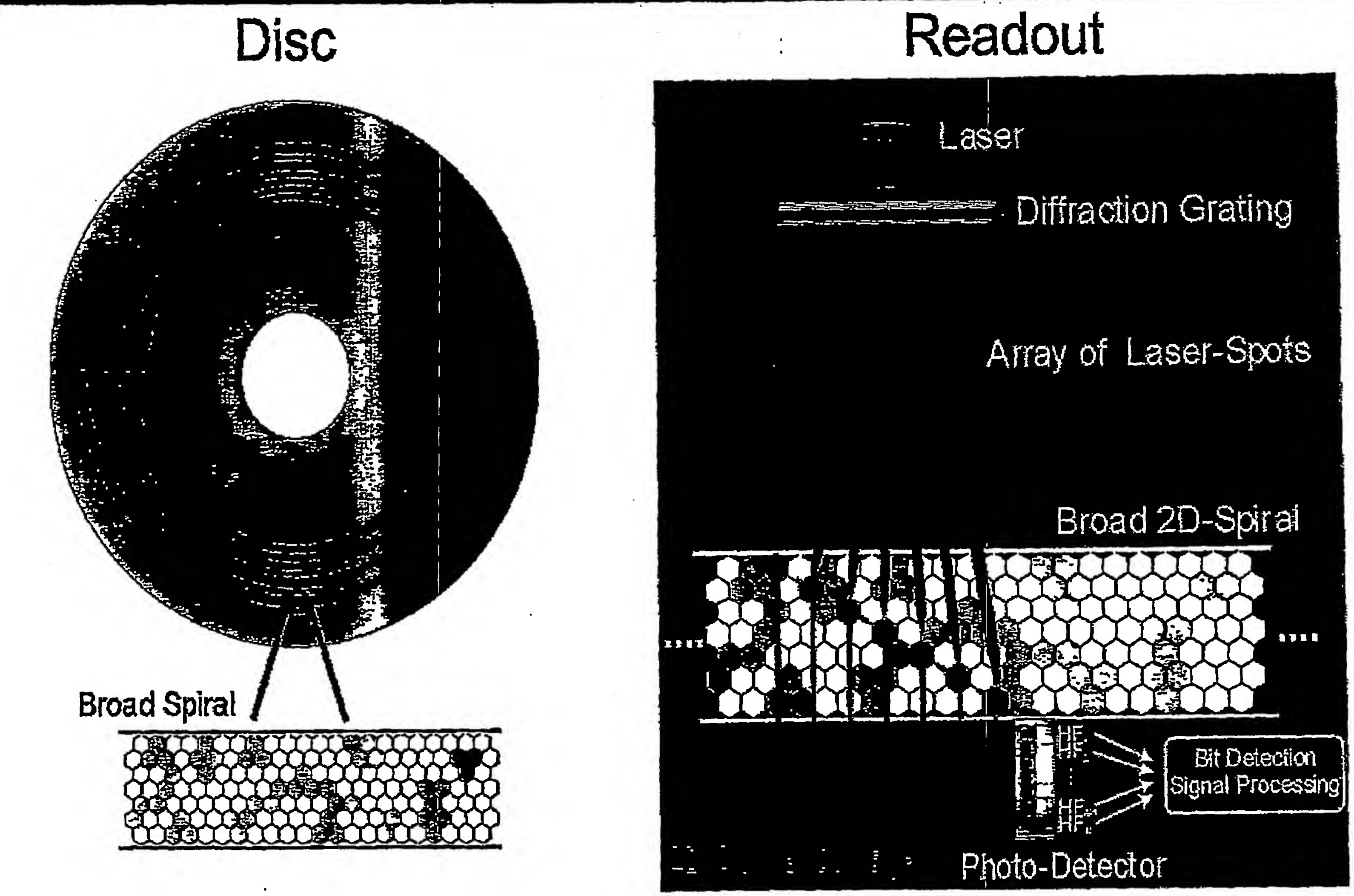



Fig. 1. TwoDOS concept.

**Philips Research** Figures on "A Catalogue of Trics for Stripe-Wise Bit-Detection" 1

**PHILIPS**

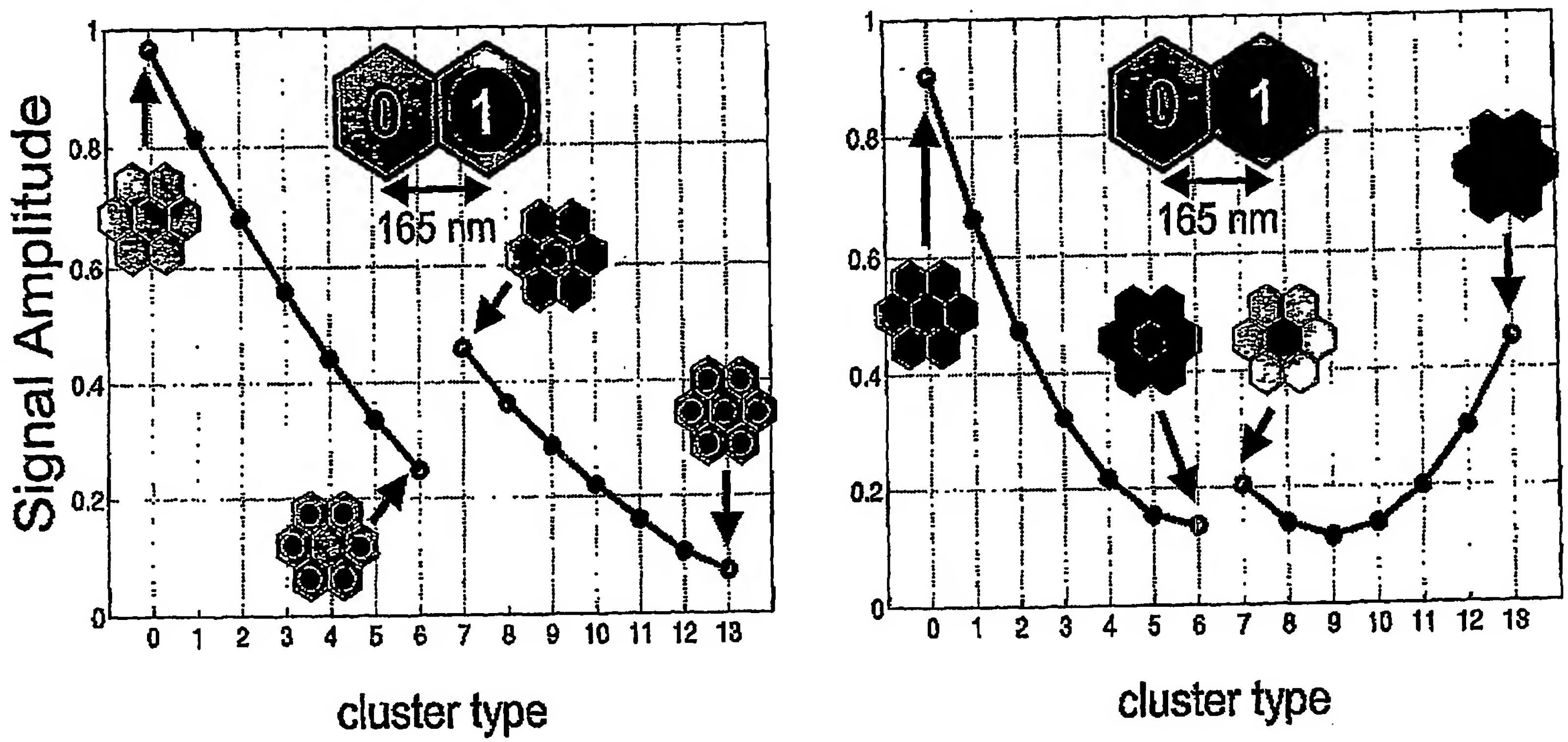



Fig. 2. 2D Optical Storage "Signal Patterns" for a capacity of 1.4x BD.
Left: optimized pit-holes with linear roll-off of the signal amplitudes.
Right: maximum-size circular pit-holes, leading to signal folding.

PHILIPS

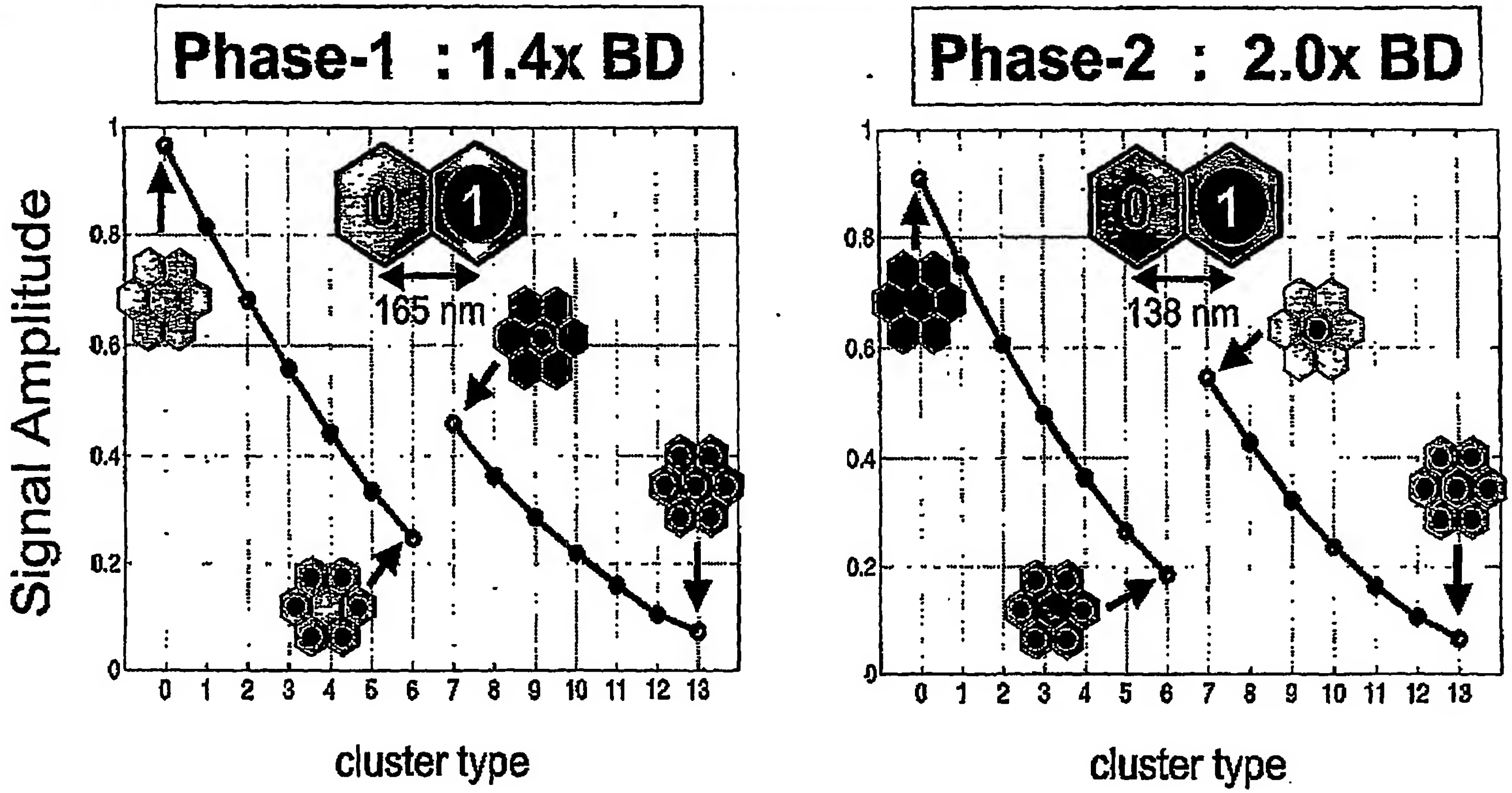



Fig. 3. 2D Optical Storage "Signal Patterns".
Left: Capacity increase of factor 1.4x over BD-format.
Right: Capacity increase of factor 2.0x over BD-format.

PHILIPS

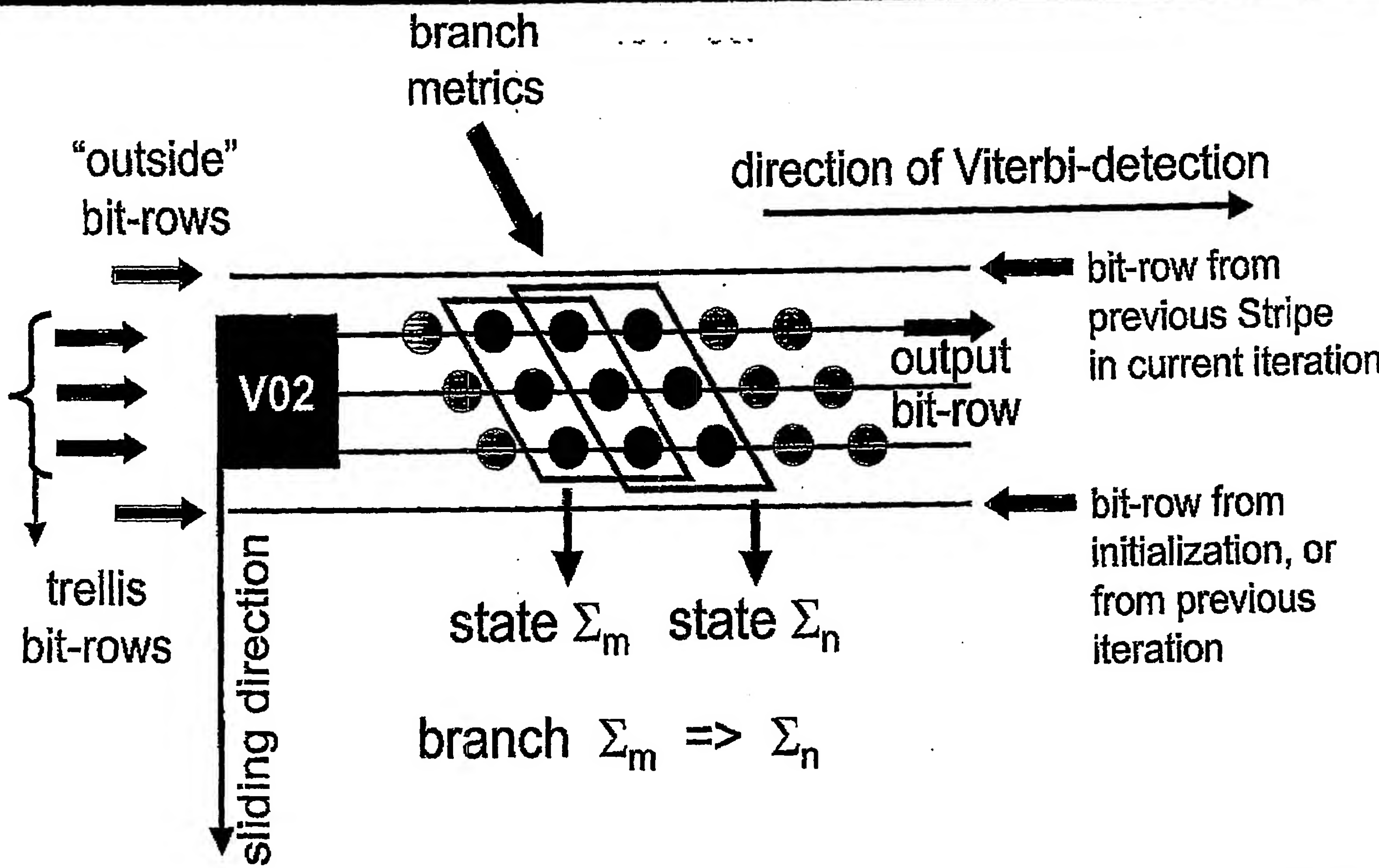



Fig. 4. States and branches for a Viterbi-detector in a 3-row stripe denoted by "V02".

PHILIPS

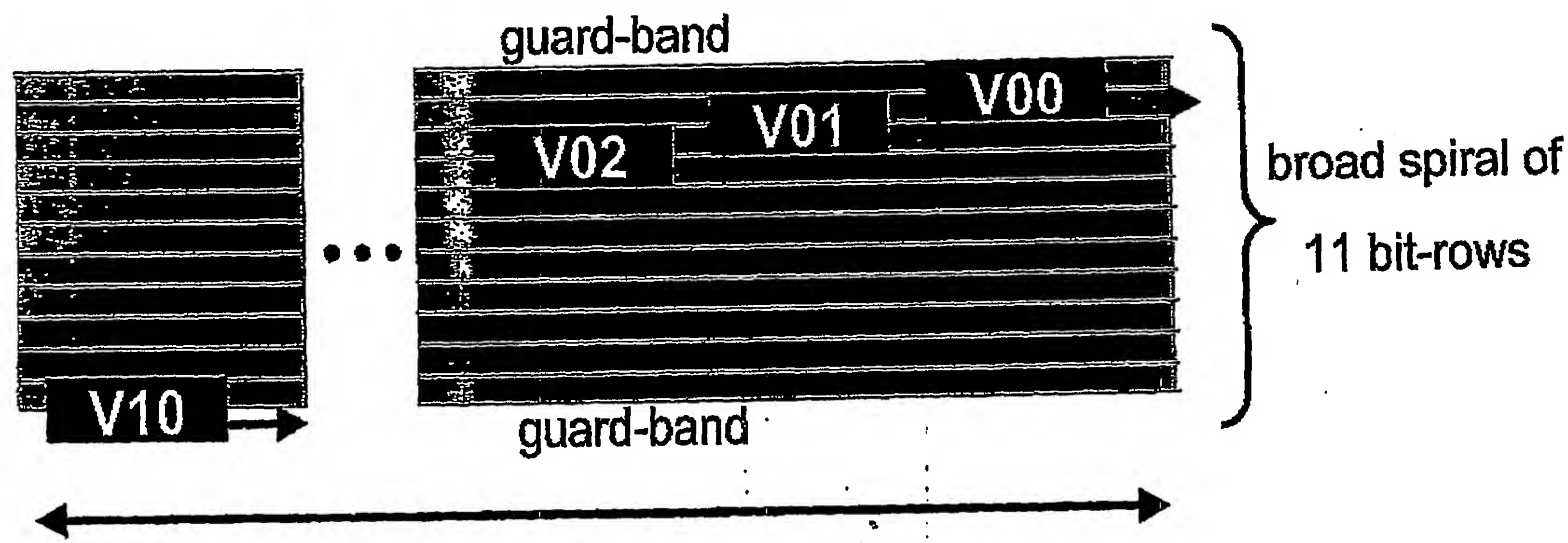



Fig. 5. "Stripe-Wise" bit-detector in a 11-row broad spiral with two guard-bands at its edges, for the case of 2-row stripes "V00", "V01", ..., "V10" cascaded one after the other with mutual delay, from top to bottom of the stripe.

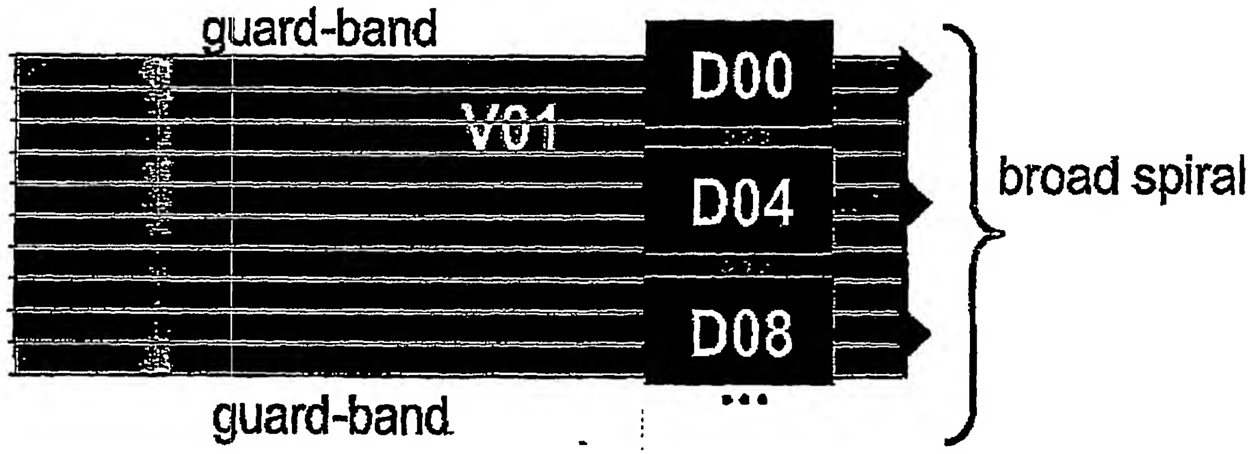



Fig. 6. Alternative bit-detection scheme (from literature) with independent 3-row Viterbi bit-detectors ("D00", ..., "D10"), each outputting their central bit-row.

PHILIPS

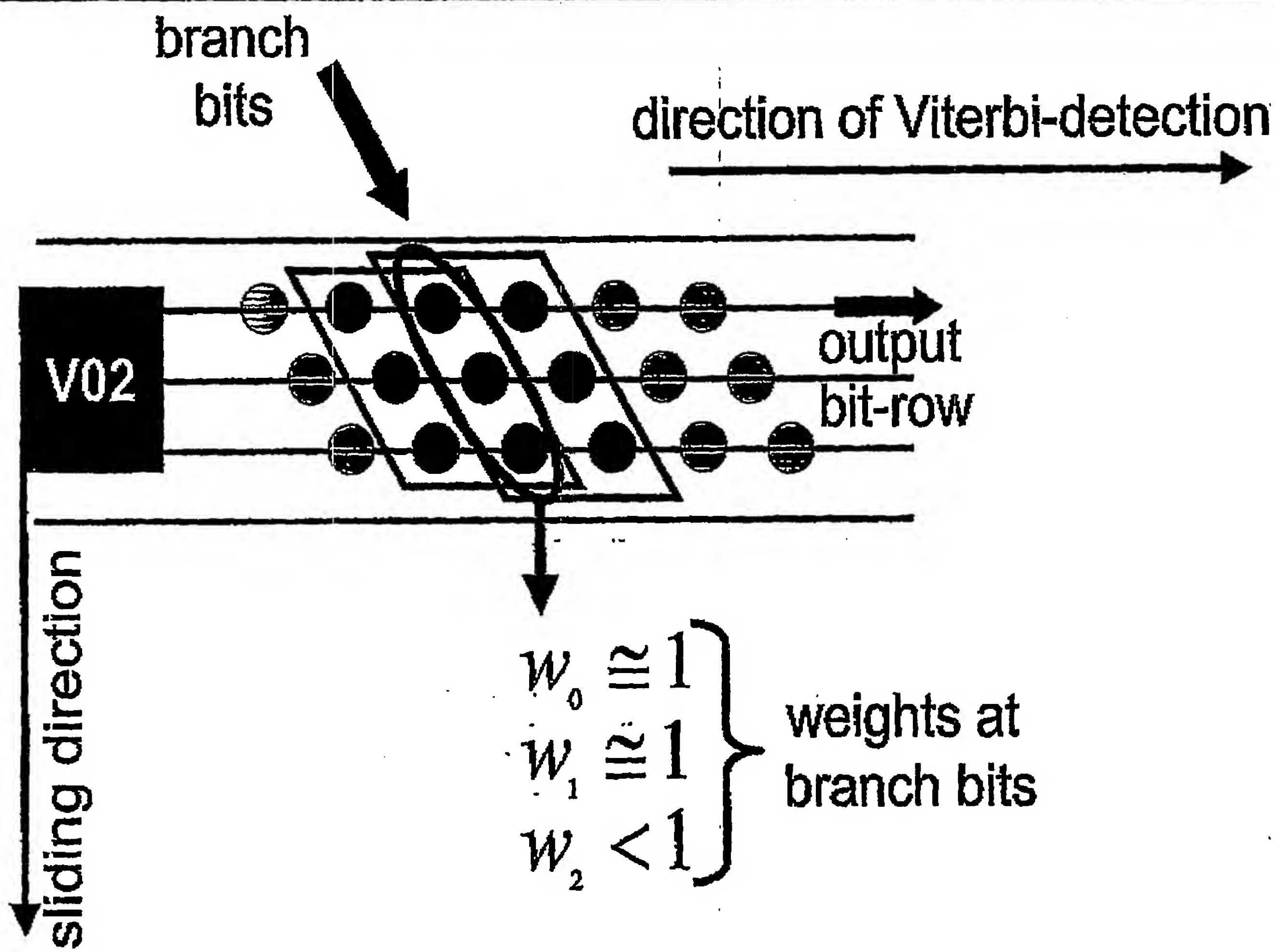



Fig. 7. Reduction of weights $w_i$ in bottom row of samples of signal waveforms contributing to branch metrics.

PHILIPS

Abstracted 2D Impulse-Response
for 2D Linear Channel @ 2x BD Capacity

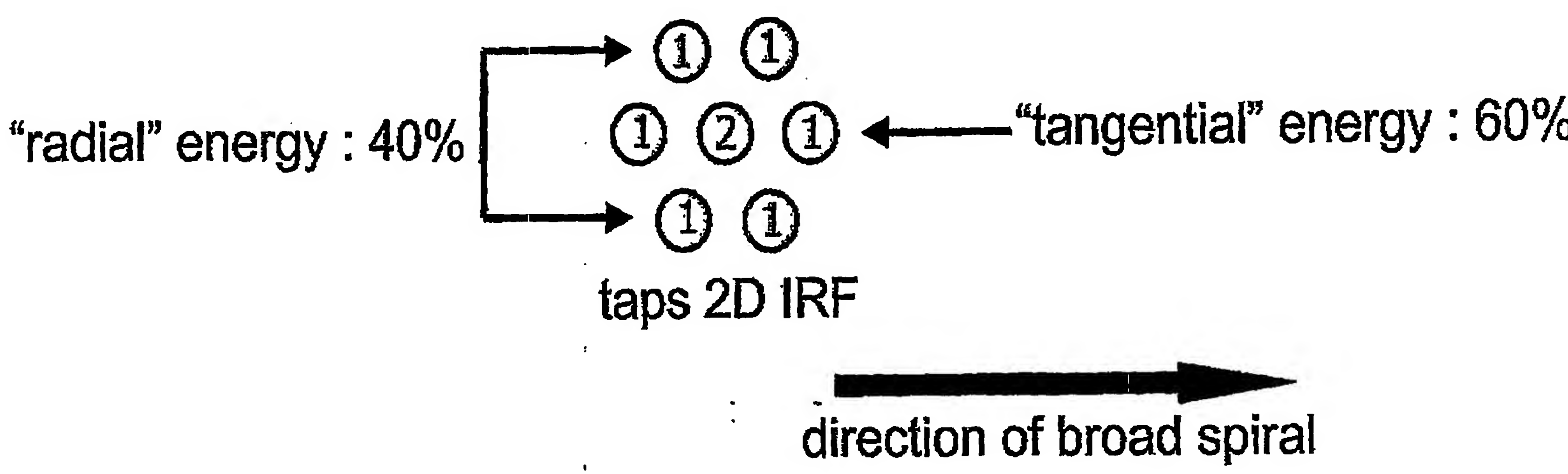



Fig. 8. 2D impulse response for high-density 2D optical storage, and the leaked-away signal-energy per bit, split in tangential and radial contributions.

**PHILIPS**

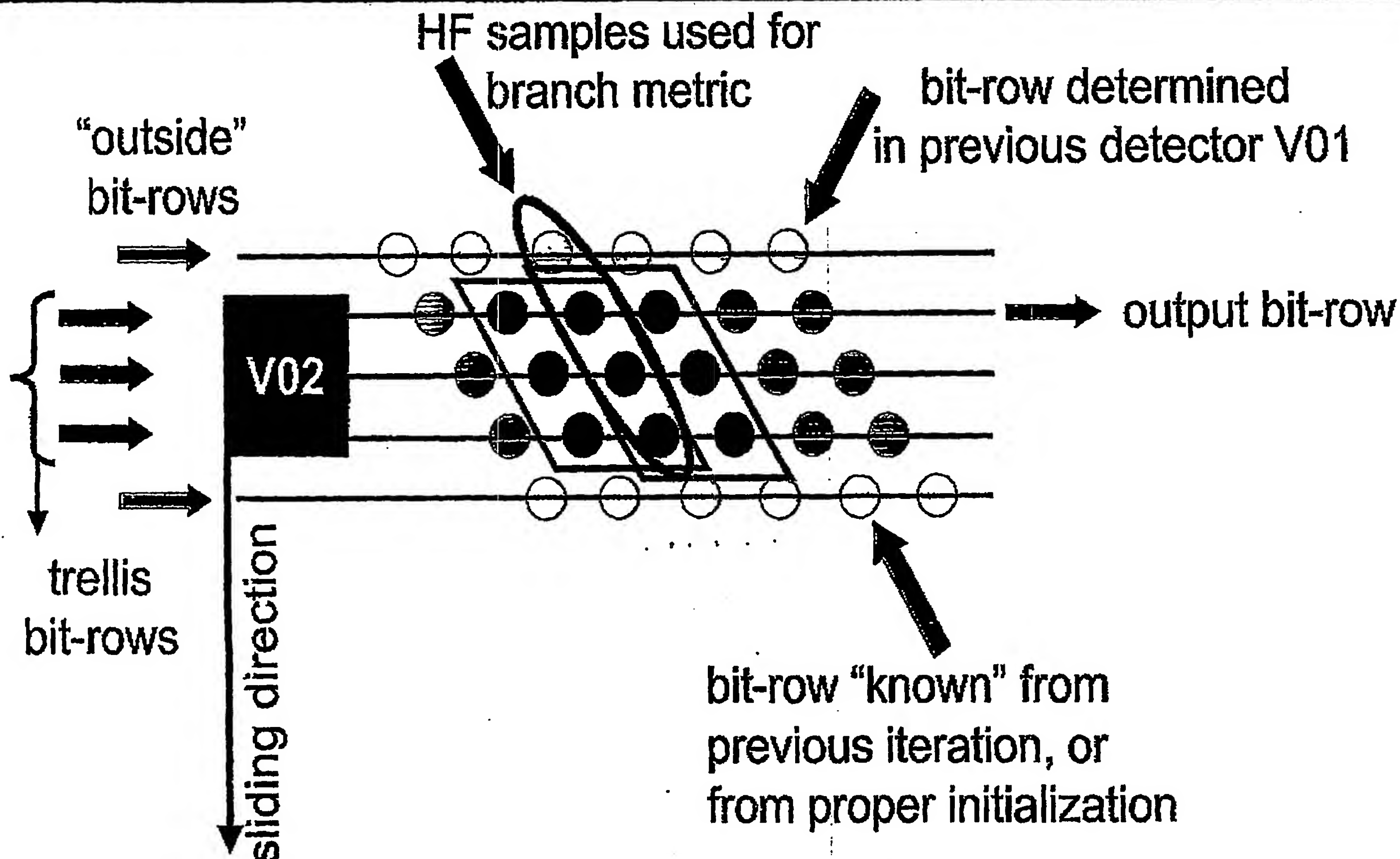



Fig. 9. Extension of computation of branch metrics with samples of signal waveform at bits in the bit-row above the stripe (assuming that the top bit-row is the output of the stripe), hereby accounting for the leakage of the signal-energy outside the stripe.

PHILIPS

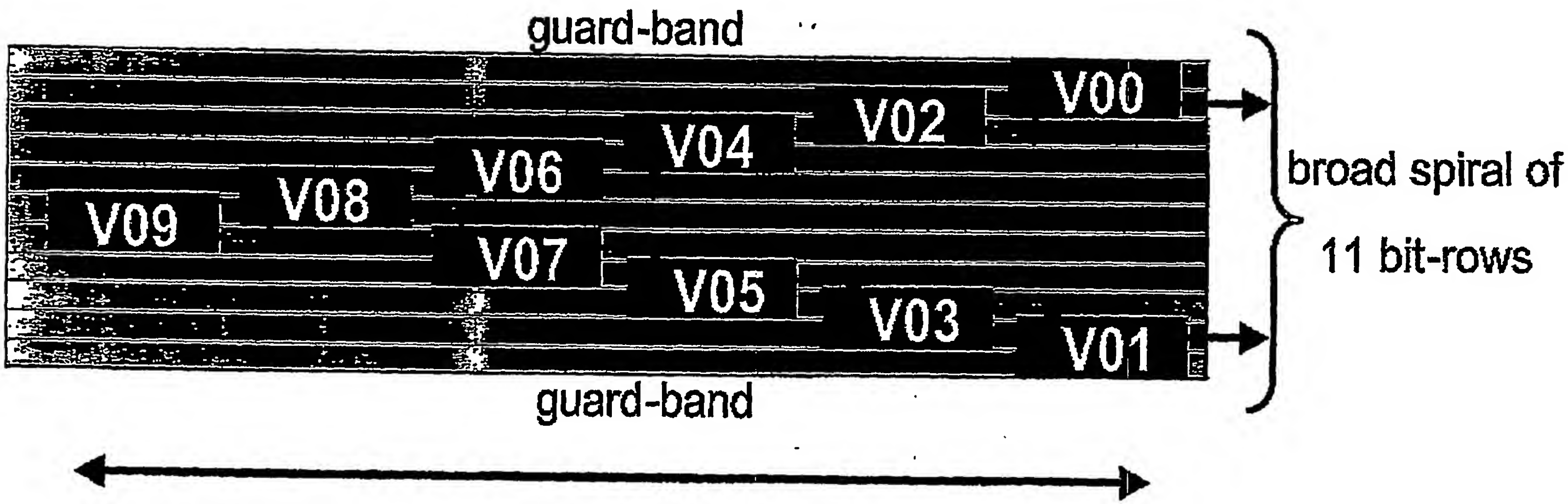



Fig. 10. V-shaped Stripe-Wise bit-detector in a 11-row broad spiral with two guard-bands at its edges, for the case of 2-row stripes. "V00", "V02", ..., "V08" are cascaded one after the other with mutual delay starting from the top guard band. "V01", "V03", ..., "V07" are cascaded one after the other starting from the bottom guard band.

**PHILIPS**

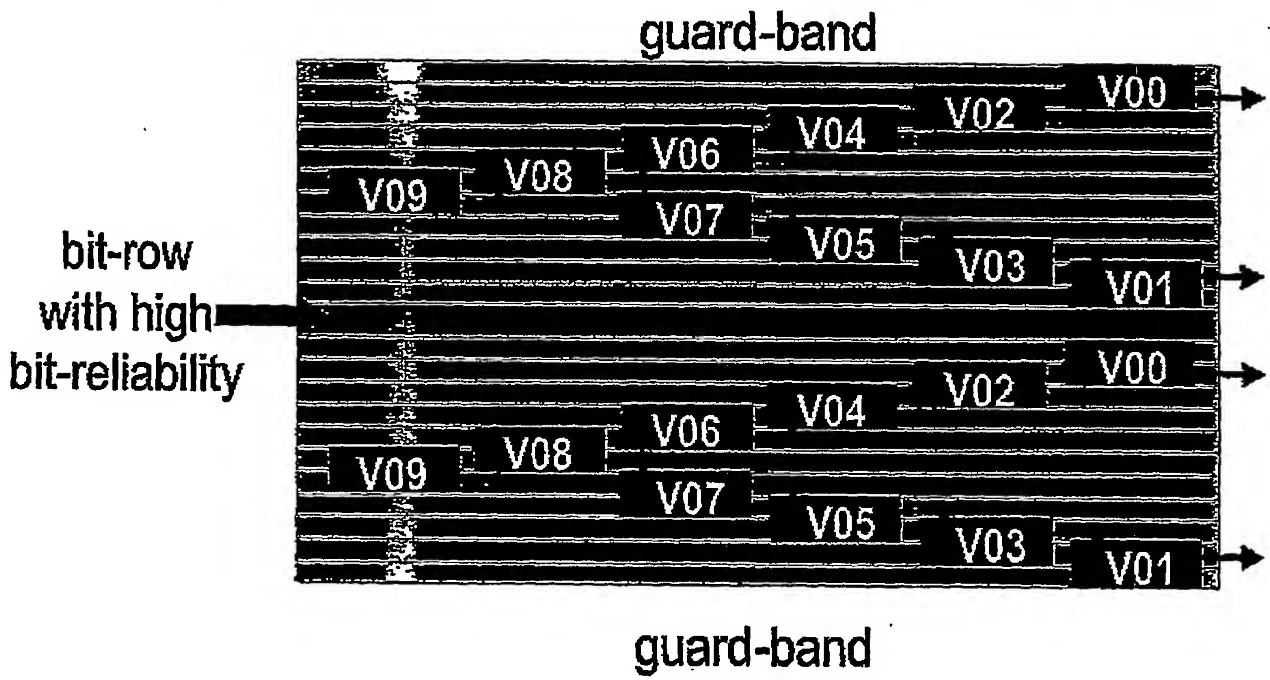



Fig. 11. Two V-shaped Stripe-Wise bit-detectors in a 23-row broad spiral with two guard-bands at its edges, and one an additional central bit-row with high bit-reliability, from which the V-shaped cascade of stripes can start propagating towards the middle row of both half-sided parts of the broad spiral. (The separate bit-detector for the reliable central bit-row is not explicitly shown).

PHILIPS

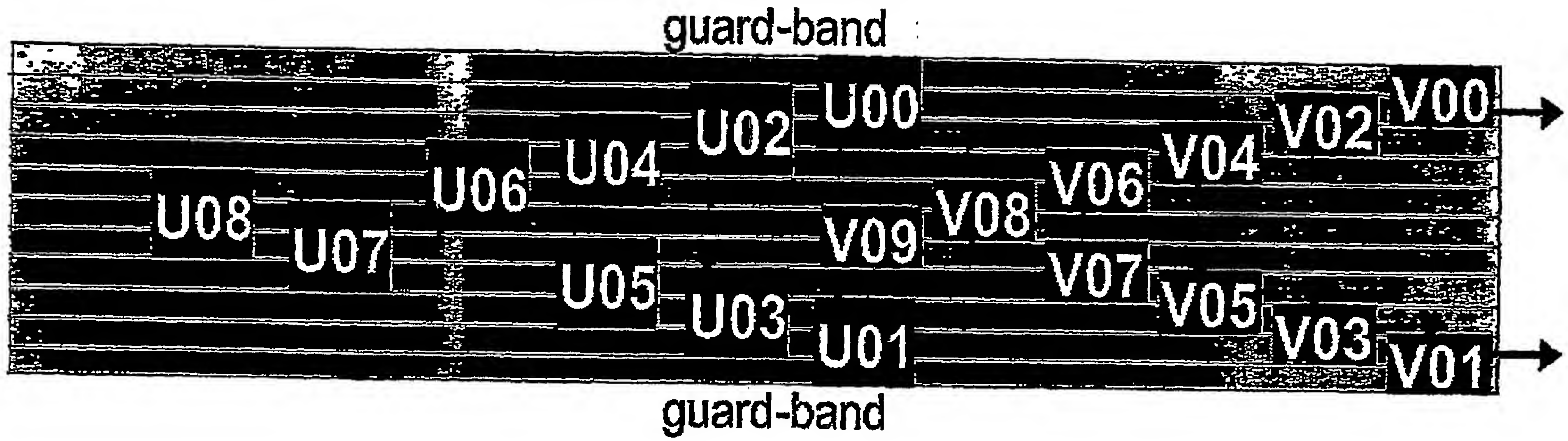



Fig. 12. Two successive iterations of V-shaped Stripe-Wise bit-detector in a 11-row broad spiral with two guard-bands at its edges. The 1st iteration uses stripes of 2 bit-rows high, with the Viterbi-detectors V00, V01, ..., V09. The 2nd iteration uses stripes of 3 bit-rows high, with the Viterbi-detectors U00, U01, ..., U08.

PHILIPS

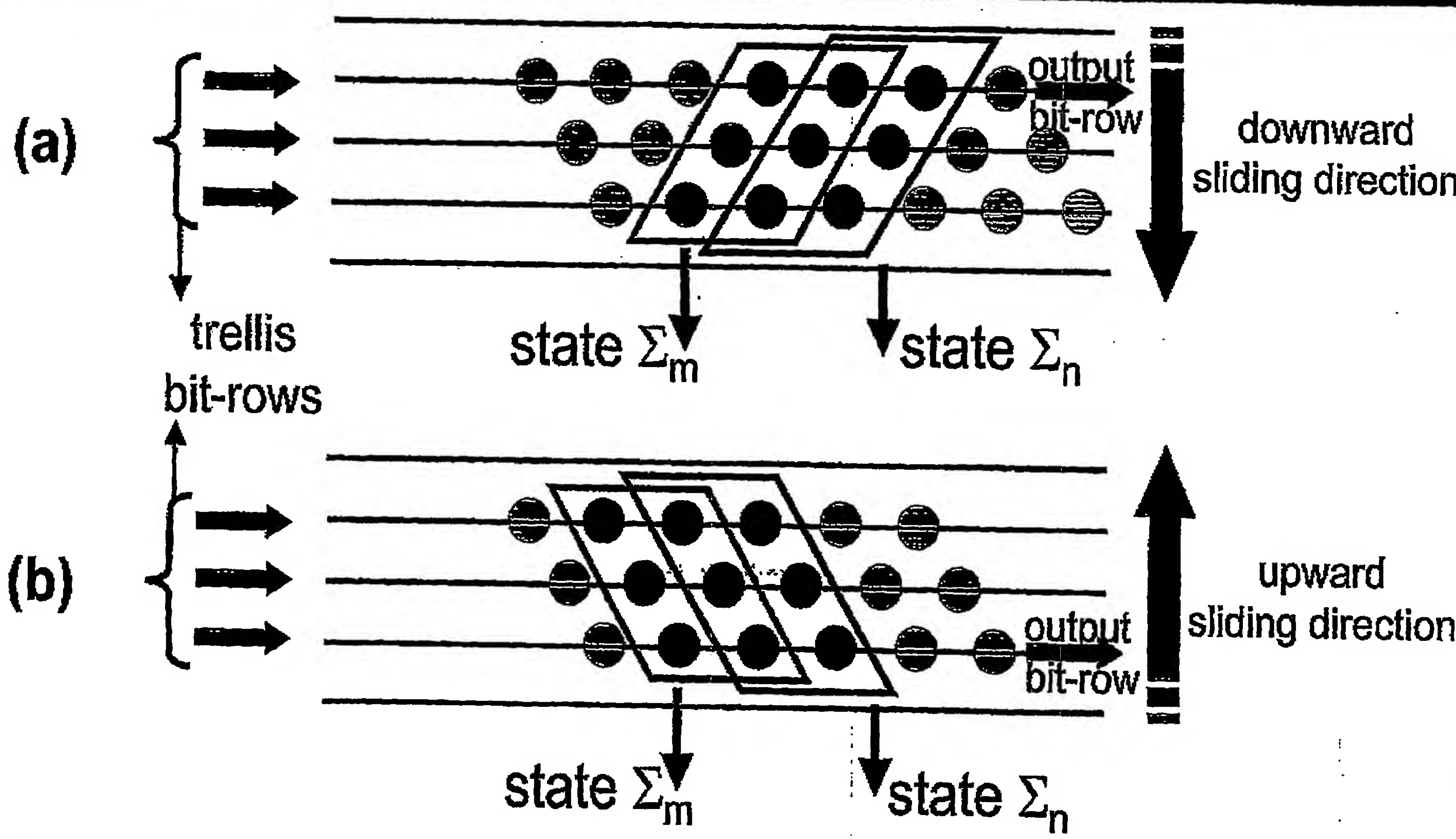



Fig. 13. State-representation for Viterbi detector in 3-row stripe. (a): the stripe is sliding from the top of the broad spiral to the center; (b): the sliding direction is from the bottom of the broad spiral to the center. Six bits are allocated to each state.

PHILIPS

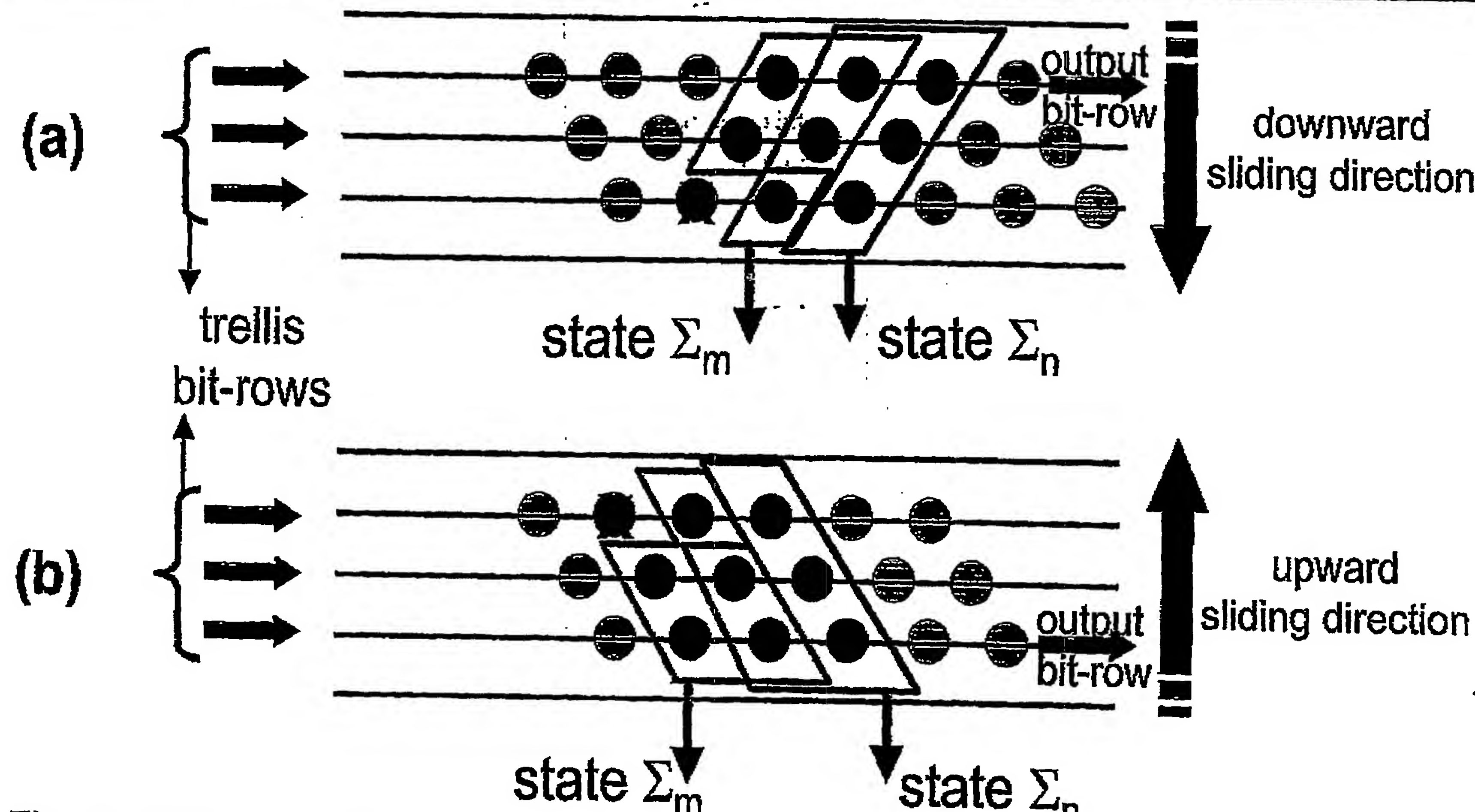



Fig. 14. Reduced trellis complexity by allocation of only 5 bits to a state for a 3-row stripe. The "missing" bit in a branch is indicated with "X". It is needed for computation of branch metrics, and is obtained through "local sequence feedback" in the tangential direction of the spiral.

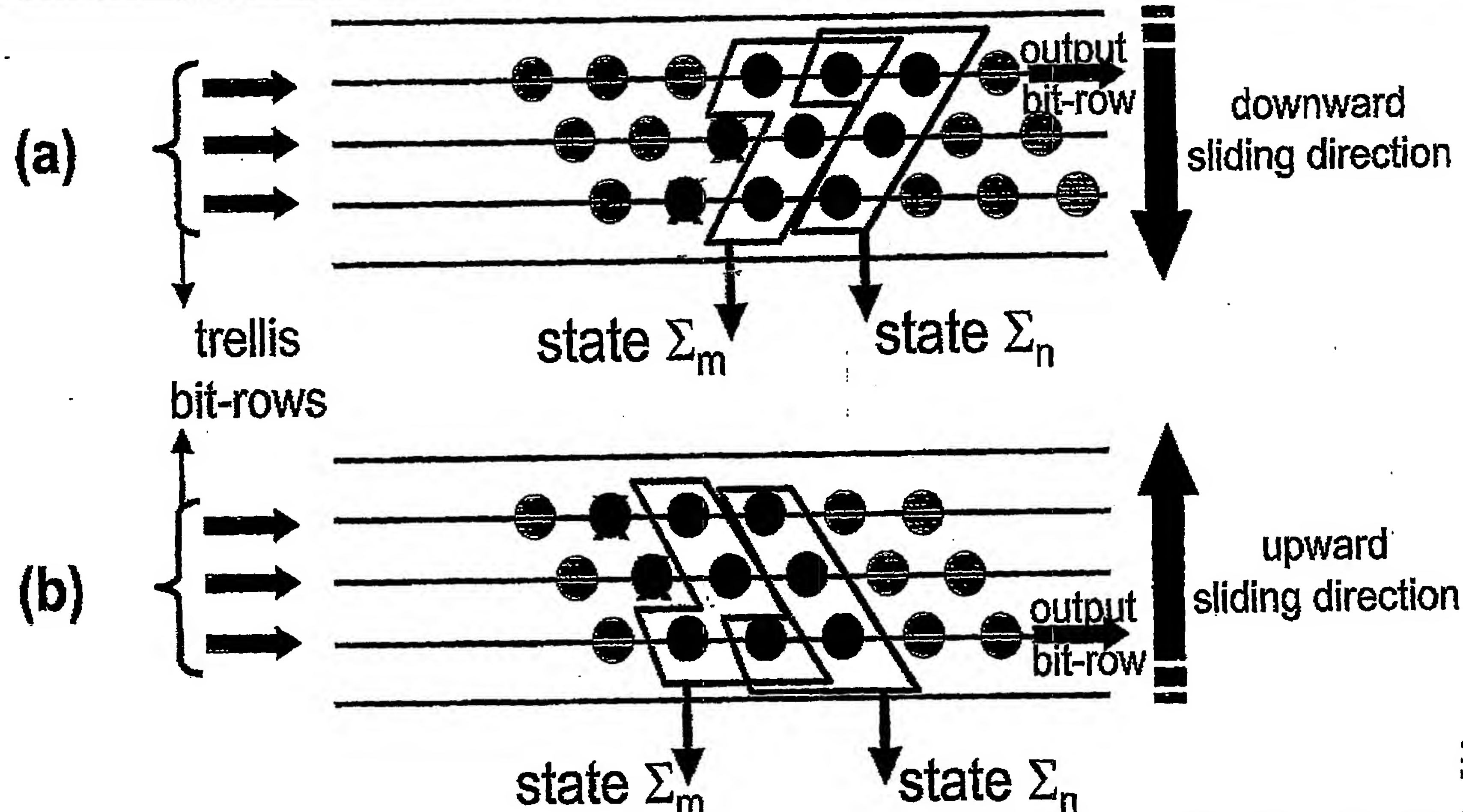



Fig. 14bis. Reduced trellis complexity by allocation of only 4 bits to a state for a 3-row stripe. The "missing" bits in a branch are indicated with "X". These are needed for computation of branch metrics, and are obtained through "local sequence feedback" in the tangential direction of the spiral.

PHILIPS

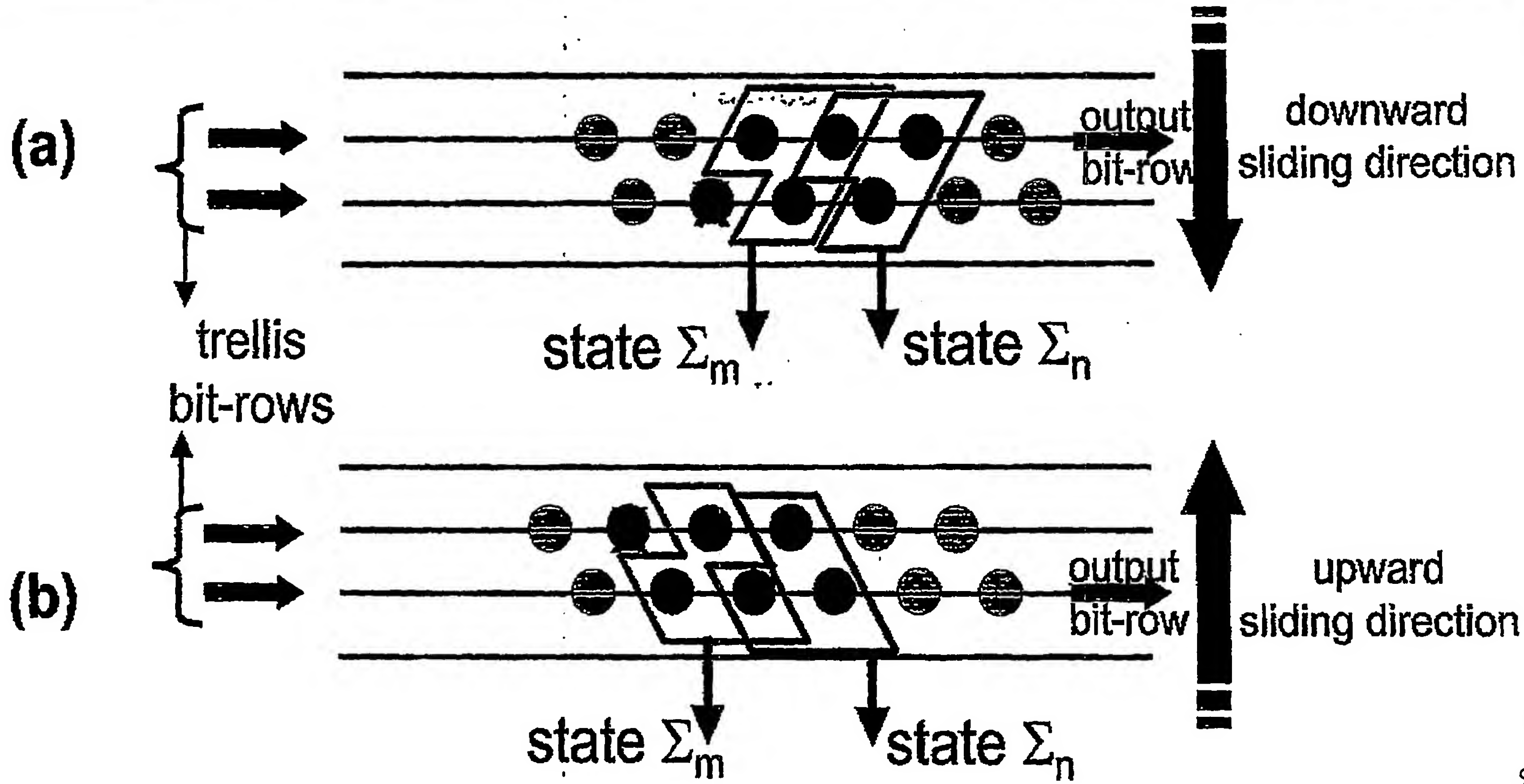



Fig. 14tris. Reduced trellis complexity by allocation of only 3 bits to a state for a 2-row stripe. The "missing" bit in a branch is indicated with "X". It is needed for computation of branch metrics, and is obtained through "local sequence feedback" in the tangentia direction of the spiral.

PHILIPS

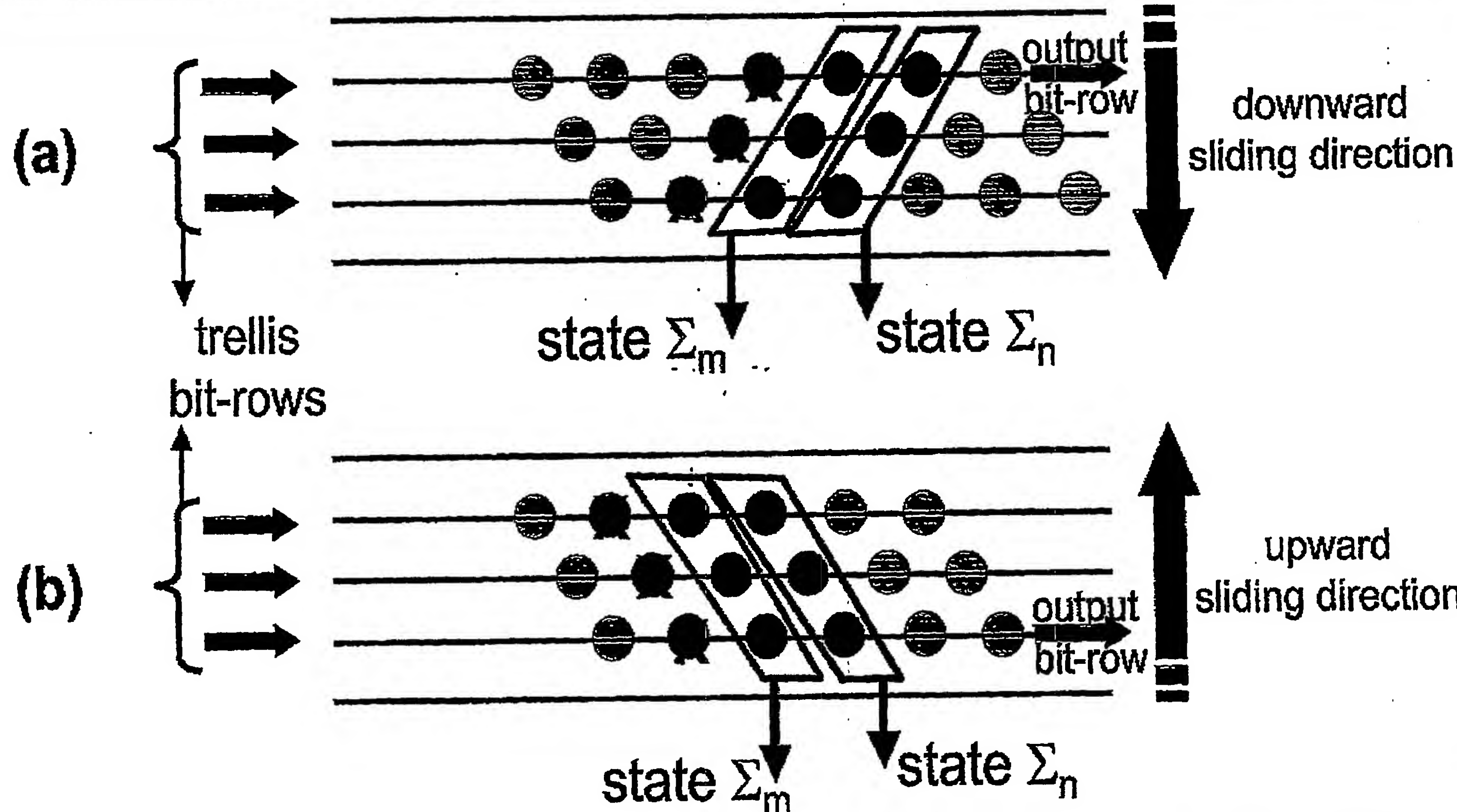
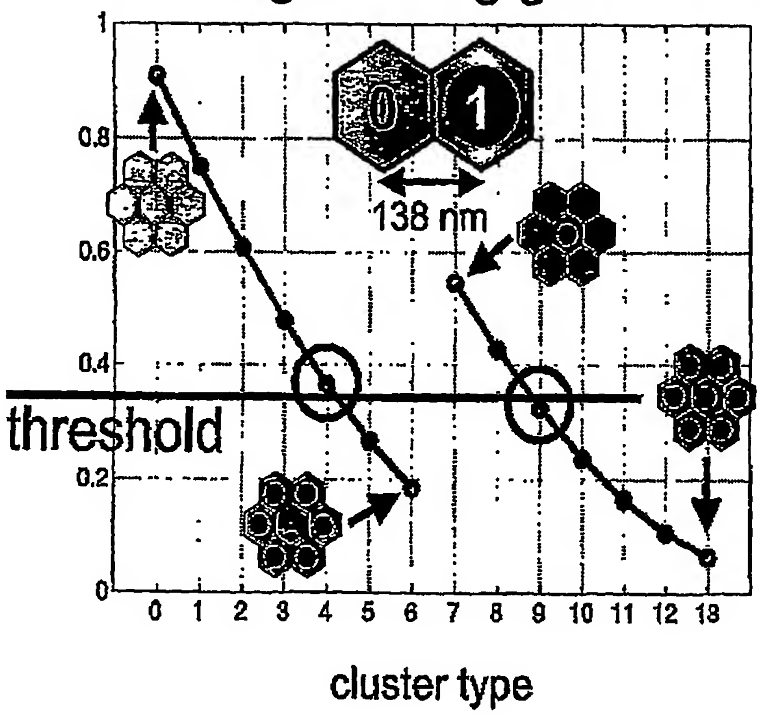
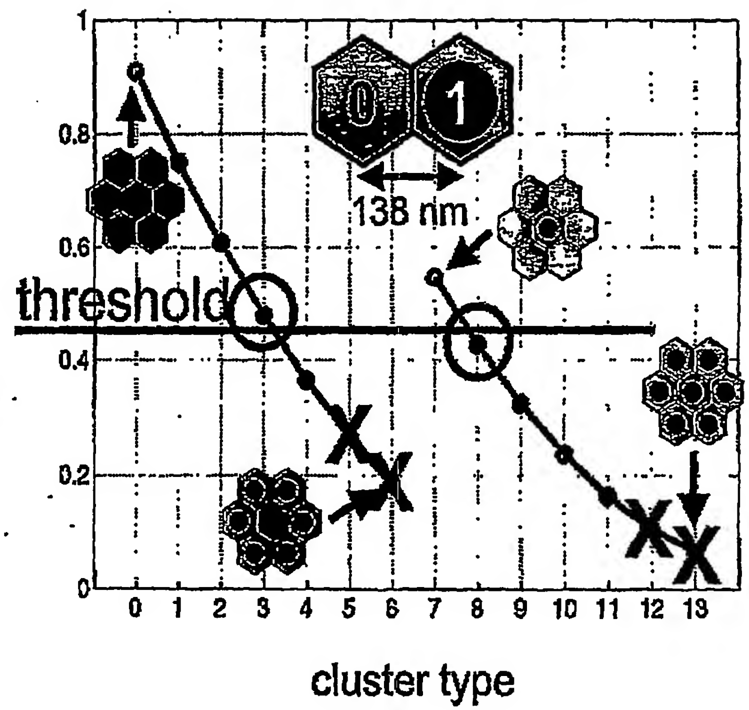



Fig. 14"". Reduced trellis complexity by allocation of only 3 bits to a state for a 3-row stripe. The "missing" bits in a branch are indicated with "X". These are needed for computation of branch metrics, and are obtained through "local sequence feedback" in the tangential direction of the spiral.

PHILIPS

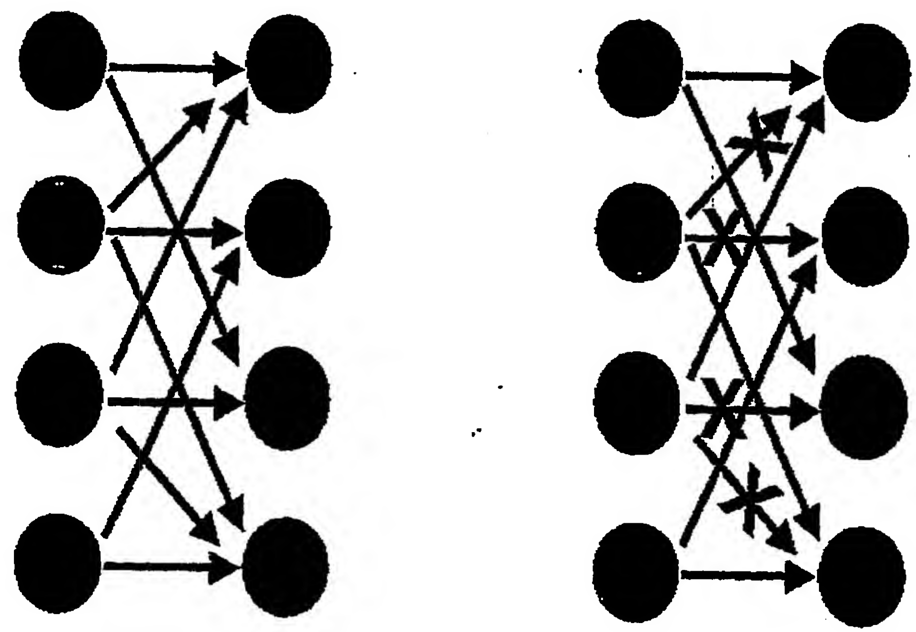



○ : cluster-level used to derive threshold level

Fig. 15. Threshold-levels for preliminary initial bit-decision through threshold-detection (or slicing). The "X"'es represent cluster-levels that cannot occur in the bit-rows that neighbour the guard-band.

PHILIPS



(a) (b)

Fig. 16. Basic Principle of "Reduced State Techniques". Left (a): original trellis. Right (b): Trellis with eliminated branches for some of the states.

PHILIPS

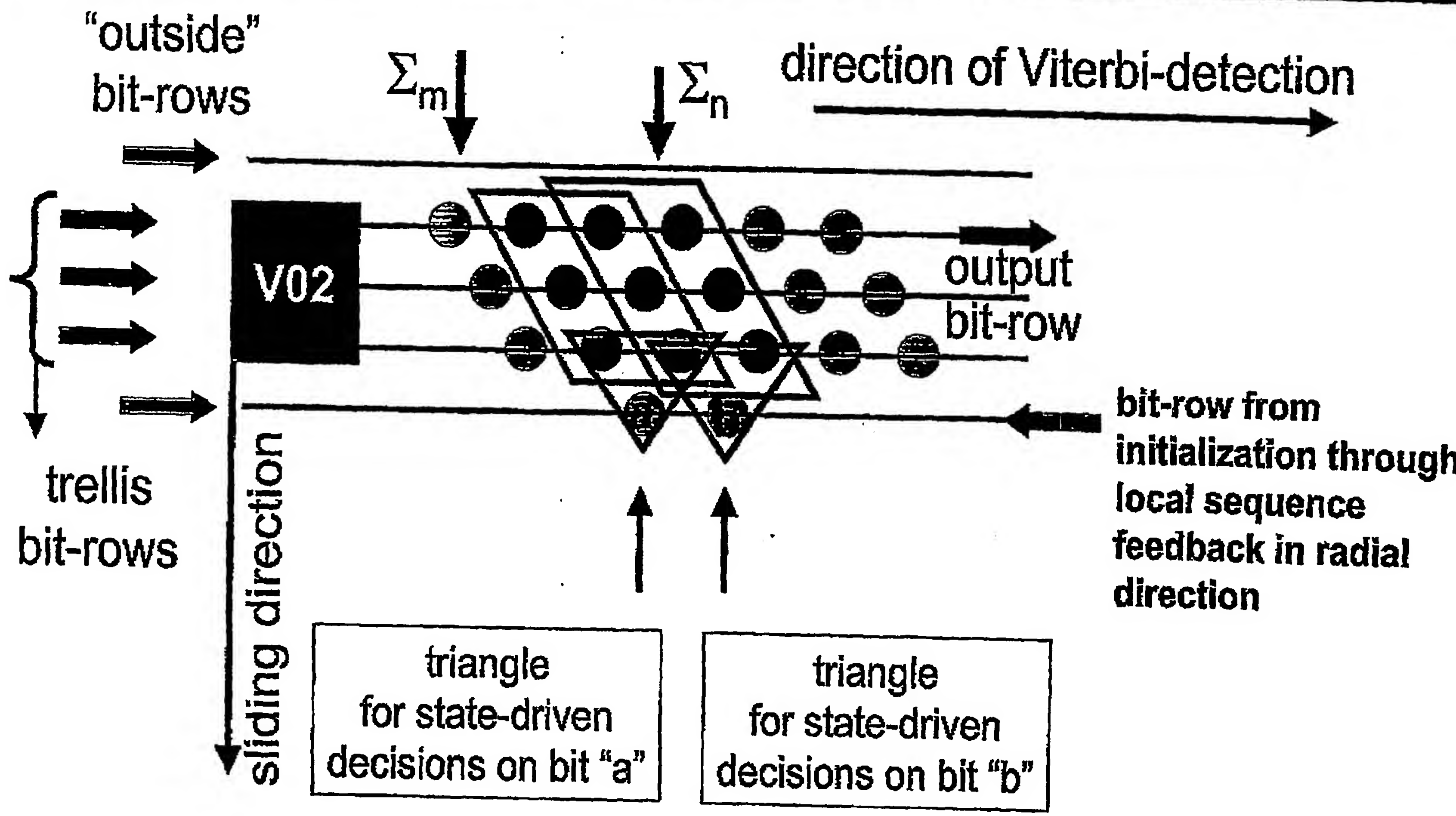



Fig. 17. Principle of state-dependent initialization through "local-sequence-feedback" in radial direction for bits "a" and "b" of the neighbouring bit-row at the bottom of the stripe.

PHILIPS

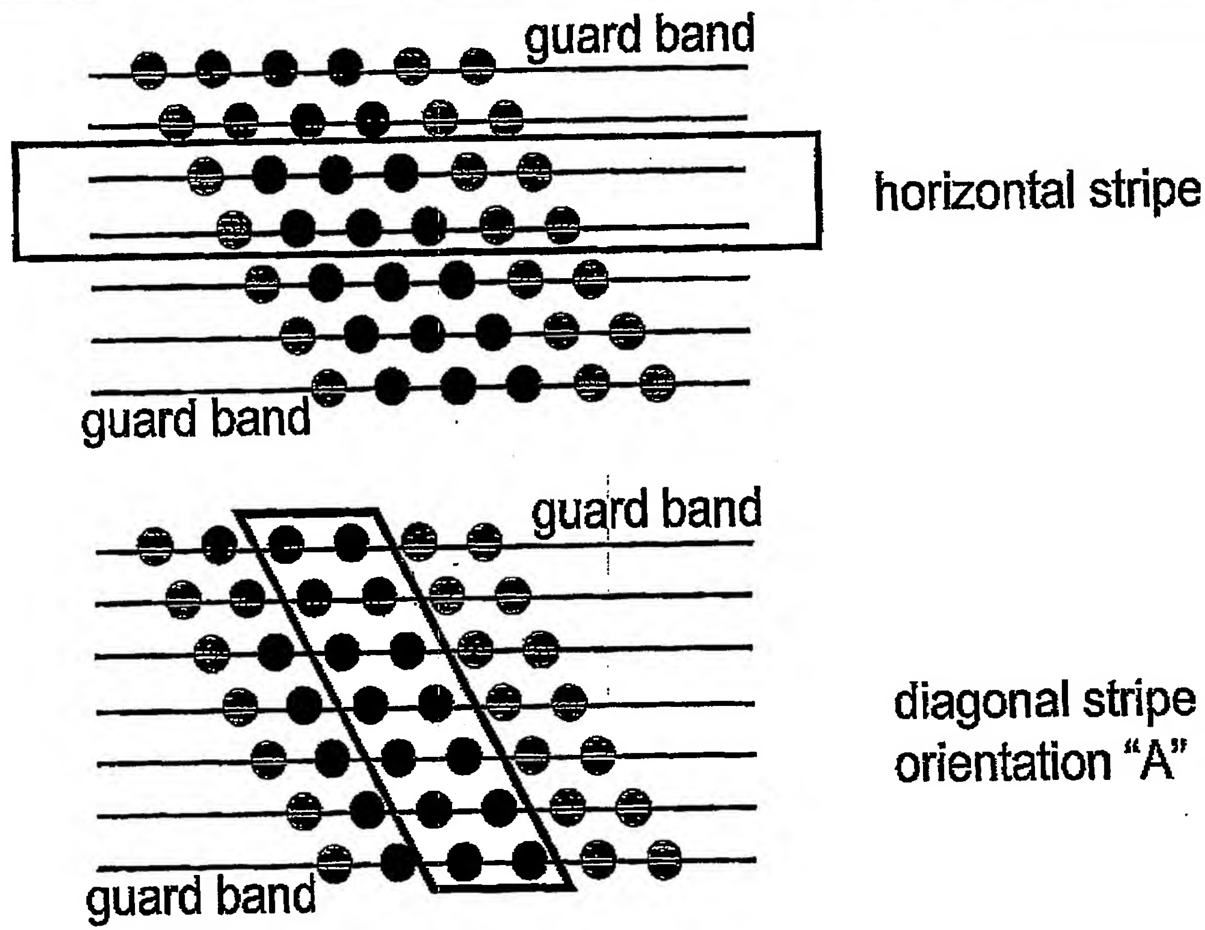
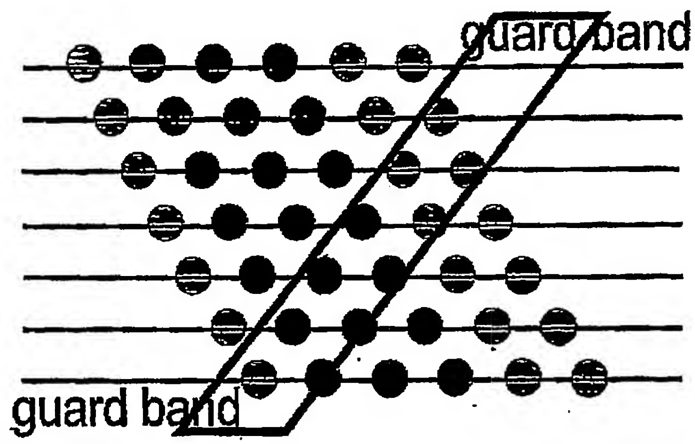



Fig. 18. Other possible orientations of a stripe (assumed 2 bits wide) in a broad spiral.

**PHILIPS**




diagonal stripe
orientation "B"

Fig. 18. Continued.